SMART JAM

Master Executive en Business Intelligence y Big Data 2014/2015

AUTORES
Ana Cantero Pozo (España)
Ignacio Bonet Cifuentes (España)
Javier Lamana Pérez (España)
Jesús Villagra Laso (España)

Fecha

Contenido

1 ANÁLISIS DEL PROBLEMA DE NEGOCIO 4
1.1 Análisis del problema 4
1.2 Información de la solución 5
1.3 Recopilación de información para la concreción de la solución 6
1.3.1 Manifestaciones 6
1.3.2 Incidentes 7
1.3.3 Presupuesto para Smart Cities 9
1.3.4 Financiación 11
2 MISIÓN, VISIÓN Y VALORES 14
2.1 Misión 14
2.2 Visión 14
2.3 Valores 14
3 ANÁLISIS DAFO 15
4 PLAN ESTRATÉGICO 16
4.1 Business Model Canvas 16
4.2 Objetivos 18
4.2.1 Descripción de la idea 18
4.2.2 ¿Qué ofrecemos? 20
4.2.3 ¿Quiénes son nuestros potenciales clientes? 23
4.2.4 ¿Cuál es nuestra ventaja competitiva? 24
4.2.5 ¿Cuál es nuestro ámbito geográfico de acción? 25
5 PLAN DE ACCIÓN 27
5.1 Alcance 27
5.2 Análisis del talento humano 27
5.3 Análisis de los recursos físicos 27
5.3.1 Arquitectura 27
5.3.1.1 Infraestructura 27
5.3.1.2 Servicios empleados 27
5.3.1.3 Estructura 32
5.4 Actividades y Tareas 33
5.4.1 Recopilación de fuentes de datos 33
5.4.1.1 Orígenes de datos 33
5.4.1.2 Web Scraping 33
5.4.1.3 Web Semántica 34
5.4.1.4 Plataforma del concurso Big Data Challenge 2015 (Telecom Italia) 34
5.4.1.5 Estructura de los datos (antes de ETL) 37
5.4.1.6 Modelo de datos 39
5.4.2 Tratamiento de las fuentes de datos 39
5.4.2.1 Carga de datos en sistema de ficheros distribuido de Amazon (S3) 39
5.4.2.2 Tratamiento de ficheros previo a ETL 40
5.4.2.3 ETL 41
5.4.2.3.1 ETL en *Hive* 41

5.4.2.3.2 ETL en ArcGIS 42
5.4.3 Proceso Predictivo 44
5.4.3.1 Comprensión del negocio 44
5.4.3.2 Comprensión de los datos 44
5.4.3.3 Preparación de los datos 45
5.4.3.4 Modelado 46
5.4.3.5 Evaluación 48
5.4.3.6 Despliegue 53
5.4.4 Desarrollo de cuadro de mando en Qlik Sense 54
6 EVOLUCIONES FUTURAS 56
7 GESTIÓN DEL TIEMPO 57
7.1 Planificación y gestión temporal del proyecto 57
7.1.1 Definición del problema 57
7.1.2 Recopilación de información 58
7.1.3 Desarrollo del piloto 59
7.1.4 Resumen. 61
8 INDICADORES 62
9 PLAN ECONÓMICO-FINANCIERO. 63
10 ANEXOS 66
10.1 Entrevistas. 66
10.1.1 Anexo II: Entrevista con el alcalde de Palencia. 66
10.1.2 Anexo III: Entrevista con la gerente de Business Intelligence de Ferrovial 69
10.2 Anexo IV: Scripts 71
10.2.1 Obtención de tweets mediante consulta a API 71
10.2.2 Pasos previos a la ETL en Python 72
10.2.2.1 Parseo puntos de interés 72
10.2.2.2 Parseo Tweets 75
10.2.3 ETL de las fuentes no parseadas 76
10.2.4 ETL de las fuentes parseadas 78
10.2.5 ETL creación de QVDs Qlikview 81
10.2.6 Word Count 83
10.2.7 Análisis predictivos 84
10.2.7.1 preparacionRoma.py 84
10.2.7.2 modeladoRoma.py 86
10.2.7.3 evaluacionRoma.py 88
10.2.8 Scraping 90

1 ANÁLISIS DEL PROBLEMA DE NEGOCIO

1.1 Análisis del problema

El problema a resolver consiste en la dificultad que existe en la actualidad para prever la aparición de aglomeraciones y concentraciones no controladas y las consecuencias que estas pueden ocasionar como la creación de avalanchas, atascos, incidentes, etc.

En las grandes ciudades, debido al gran volumen de personas y eventos (y datos que estos generan) es difícil prever la respuesta o comportamiento de los ciudadanos ante eventos y acontecimientos no planificados. Esto hace que a veces se formen aglomeraciones en lugares no habilitados o preparados para ellas y, en consecuencia, puede llegar a ser fuente de riesgos para la salud de las personas, para la conservación de los lugares y en general para el buen funcionamiento de la ciudad.

Este problema puede afectar, en mayor o menor medida, a un gran número de ciudadanos aunque según se ha observado se da con mayor facilidad cuando se da la combinación de estos tres factores: gran ciudad, evento multitudinario y recintos no adecuados.

Tanto las instituciones públicas como las empresas privadas han percibido el problema y también la dificultad de controlarlo y predecirlo. En concreto, las instituciones y organismos públicos se ven afectados debido a que son los encargados de velar por la seguridad ciudadana y, como se ha comentado, esta puede verse en peligro; respecto a los propios ciudadanos, estos padecen los riesgos e inconvenientes originados en las aglomeraciones en forma de cortes, atascos e incidentes; por último, también afecta al sector privado ya que en muchas ocasiones sus negocios se ven afectados por los desperfectos ocasionados en estas aglomeraciones descontroladas.

Lamentablemente, el problema está de actualidad ya que quedan muy recientes incidentes en fiestas y discotecas en los que murieron varias personas por aplastamiento. En una capital grande como Madrid están a la orden del día las aglomeraciones y concentraciones no controladas debido a manifestaciones, espectáculos, acontecimientos deportivos, etc.

1.2 ***Información de la solución***

Como se puede suponer se trata de un problema complejo ya que depende de varios factores que son difíciles de controlar y por el momento aún no ha sido resuelto de forma global por ninguna solución. Hay algunas iniciativas en marcha como la del gobierno de China[1] pero aún está por llegar una herramienta que permita dar una solución real al problema en todas sus vertientes. Es por ello que consideramos que la creación de una aplicación, basada en técnicas de inteligencia artificial, capaz de predecir y alertar de la posibilidad de aglomeraciones sería de gran utilidad y tendría una gran acogida. Para conseguirlo, el éxito estará muy relacionado con la calidad y cantidad de datos que manejemos y también con la forma de interpretarlos y analizarlos. Por otro lado, será de vital importancia dada la cantidad de datos y procesamiento de los mismos, los recursos computacionales a nuestra disposición. Dada la complejidad del problema si no se dan al menos estas condiciones las posibilidades de tener éxito disminuyen considerablemente.

Trataremos de contar con el respaldo o colaboración de las empresas propietarias de los datos de movilidad y comunicaciones de las personas. Éstas, tendrán interés en la herramienta para poder sacar provecho a la información generada por sus clientes y proporcionarán la materia prima que serán los datos. Entre las colaboradoras podríamos considerar a las empresas de telecomunicaciones, los bancos, empresas aseguradoras, agencias de marketing y publicidad, etc. Consideramos que podemos encontrar rechazo a la colaboración en el proyecto en aquellas compañías que estén trabajando en su propia solución al problema pero, sin embargo, otras pueden ver una oportunidad de negocio y apostar fuerte por ello ya que sería una forma de entrar en el negocio sin tener que implementar desarrollos internos, solo proporcionando los datos y con la posibilidad también de una inversión económica. Por otro lado, verían una fuente de ingresos en la explotación de la información que han proporcionado. Si se llega a acuerdos con las

[1] http://www.elcorreo.com/alava/tecnologia/investigacion/201501/26/software-evitara-avalanchas-20150126124611-rc.html

empresas propietarias de estos datos se tendrá una importante parte del camino recorrido (ya que la compra de estos datos seguramente resultaría excesivamente cara) con lo que se impone tratar de alcanzar alianzas y colaboraciones comerciales. A partir de estos datos, la herramienta se encargará de transformarlos en información y conocimiento para poder llevar a cabo las acciones adecuadas para solucionar el problema.

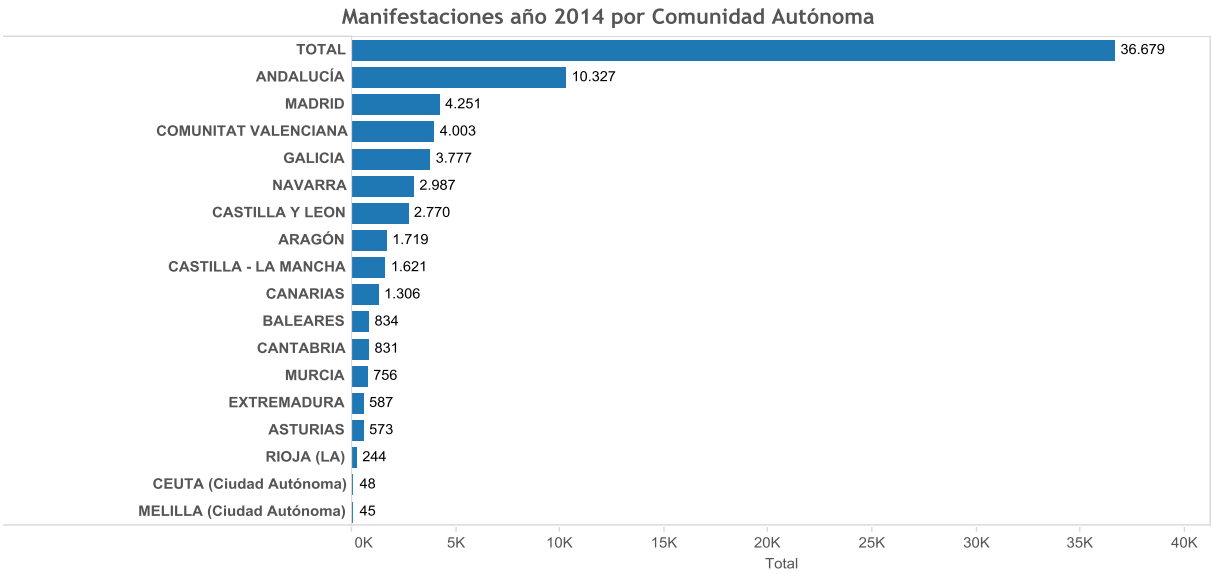
1.3 ***Recopilación de información para la concreción de la solución***

Para definir y evaluar correctamente el problema hemos obtenido varias métricas que nos ayudarán a conocerlo más en profundidad y, a su vez, nos proporcionarán información que nos será muy útil para encontrar la solución al mismo.

1.3.1 ***Manifestaciones***

En primer lugar veremos datos relativos a uno de los actos que en mayor medida es fuente de aglomeraciones y concentraciones como son las manifestaciones:

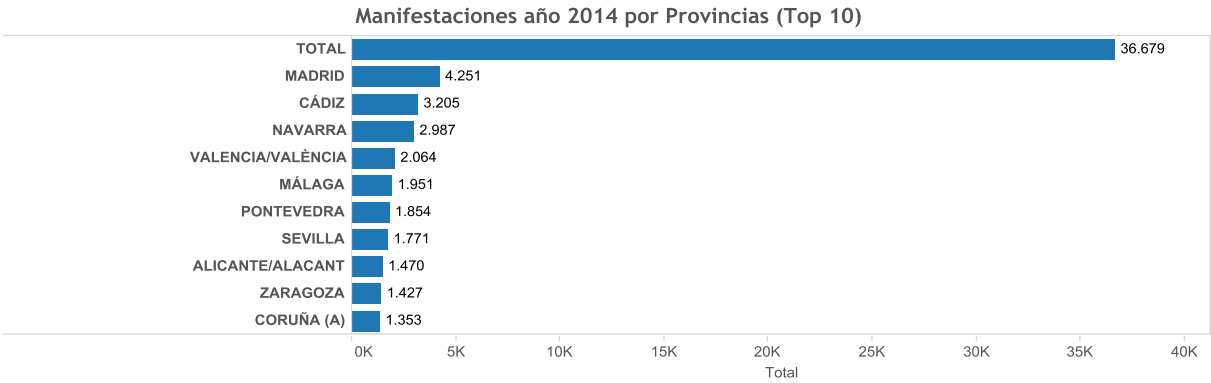
- **Manifestaciones por Comunidad Autónoma :**



Elaboración propia a partir de Datos del Ministerio de Interior.
* No se tienen datos de la Comunidad Autónoma de Cataluña.

A pesar de no tener datos de Cataluña se puede decir que las 5 o 6 primeras comunidades acumulan la mayoría de las manifestaciones que tienen lugar.

- **Manifestaciones por Provincia :**



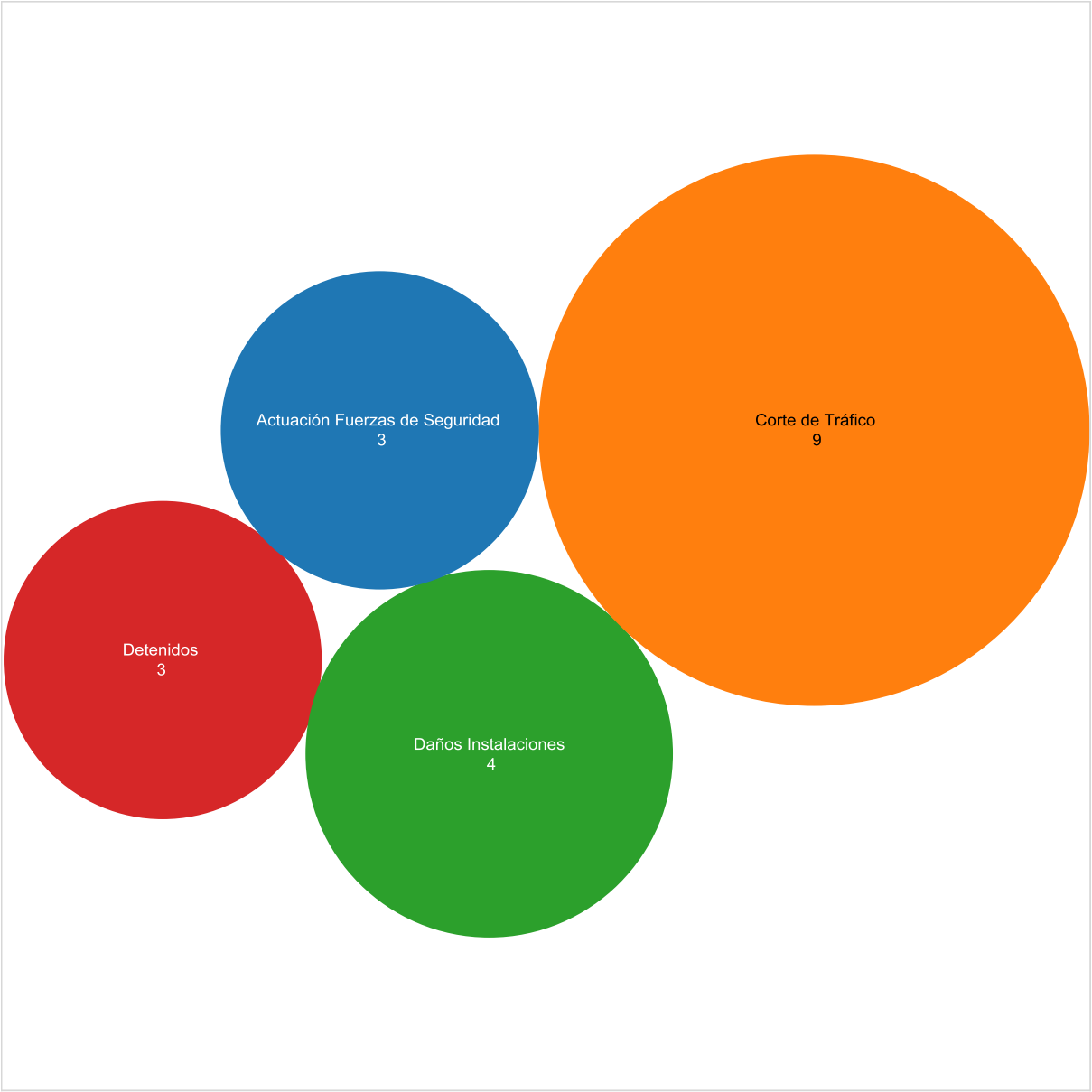
Elaboración propia a partir de Datos del Ministerio de Interior.
* No se tienen datos de las provincias pertenecientes a Cataluña.

Madrid encabeza la lista seguida en segundo y tercer lugar por provincias no tan populosas como son Cádiz y Navarra. Este dato llama la atención y nos indica que el tamaño y población de la provincia tiene menos influencia de lo que cabría esperar. También observamos que el total de manifestaciones se encuentra bastante repartido.

1.3.2 Incidentes

A continuación veremos los datos de los incidentes más comunes que se han producido en las manifestaciones. En este caso los datos pertenecen al año 2013 en la ciudad de Madrid:

Incidencias en Manifestaciones



Incidencias durante el año 2013 en la ciudad de Madrid.

Se observa que los cortes de tráfico son las incidencias que con más frecuencia se dan, casi el 50% de las veces (9 de 19). En segundo lugar los daños a las instalaciones un 20 % aproximadamente (4 de 19) y el 30% restante son las actuaciones y detenciones de las fuerzas de seguridad del estado. Un ejemplo de esto último lo tenemos en los 17 policías municipales que fueron heridos de distinta consideración en las manifestaciones del 22-M[2].

1.3.3 Presupuesto para Smart Cities

A modo de ejemplo veremos los fondos FEDER que pone Europa para el desarrollo de las ciudades e inversión en tecnología en innovación. Concretamente, los fondos I+D+i (80.000 millones de euros) para desarrollos tecnológicos y la partida de Smart Cities que tiene presupuesto de 362 millones[3]. Los proyectos suelen estar compuestos por una gran empresa que lo lidera, otras más pequeñas que también participan y, por último, las universidades. Por ciudades:

1. Burgos: gestión de aguas, 1 millón.
2. Cáceres: rehabilitación de un edificio, 4,6 millones.
3. Badajoz: eficiencia energética, 6 millones.
4. A Coruña: proyecto integral, 12 millones.
5. Málaga, también eficiencia energética, 54 millones. A los contratos que lideran Indra(A Coruña, Gijón, Rivas Vaciamadrid, Torrejón de Ardoz) y Endesa (Almería y Málaga) se suman IBM y Telefónica.
6. Madrid adjudicó el pasado mes de julio 14,7 millones a la primera (a ella sola y sin consorcio, porque la financiación no es europea).
7. Barcelona, donde se celebra el gran evento mundial de *Smart Cities*, 2 millones.

---

[2] http://www.rtve.es/noticias/20140326/ayuntamiento-madrid-cifra-166000-euros-danos-disturbios-del-22m/904341.shtml
[3] http://www.eldiario.es/hojaderouter/tecnologia/smart-city-espana-negocio-economia-ayuntamientos-empresas_0_355915190.html

8. Telefónica tiene a Ponferrada, Santander (junto a Málaga, otro de los grandes, financiado con 8,7 millones europeos) y se ha hecho recientemente con Valencia (4 millones). También con el liderazgo de un megaproyecto llamado *Fiware* ("plataforma europea para la internet del futuro"), que cuenta con 300 millones de la Unión Europea.

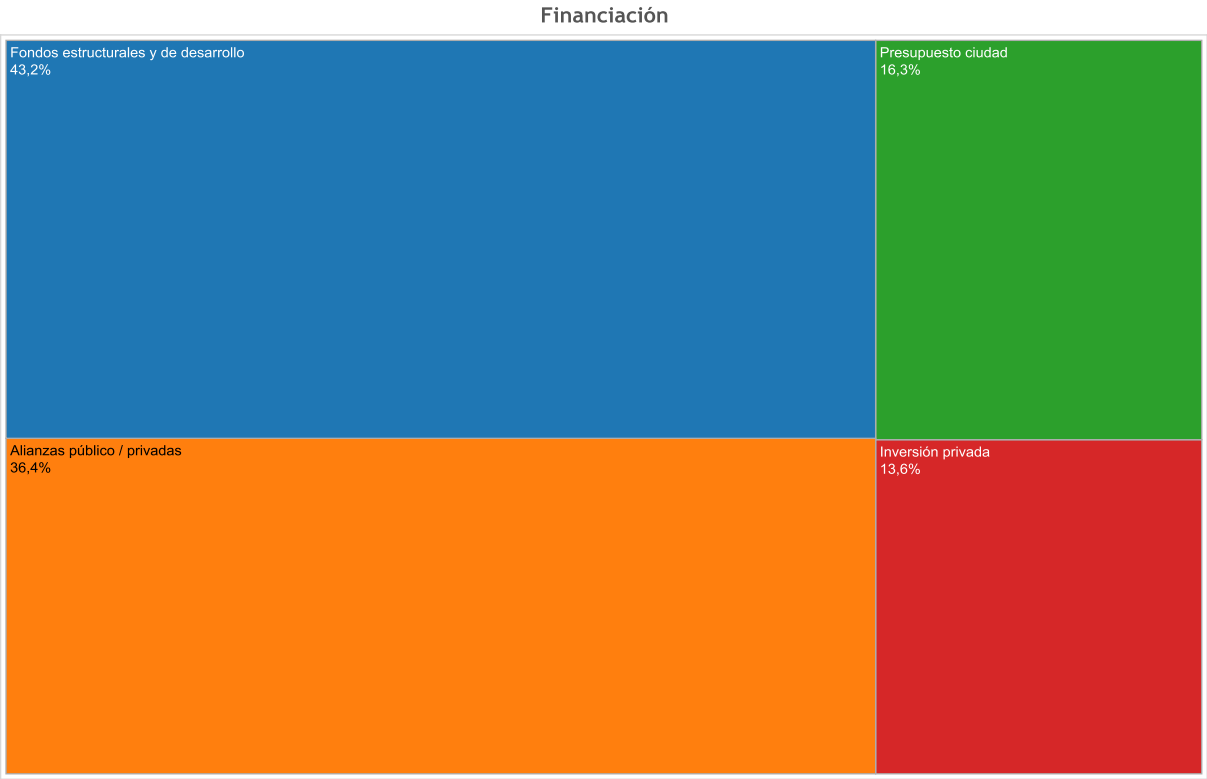
Otra muestra de la importancia que se le da por parte de los ayuntamientos a la inversión en Smart Cities (y prevenir accidentes en aglomeraciones) es el acuerdo que llegó el antiguo gobierno de Madrid para la compra de 6 drones para vigilancia de eventos multitudinarios por valor de 200.000 euros. Aunque en primera instancia el nuevo ayuntamiento iba a continuar con ello, finalmente parece que se paralizará (y se intentará modificar las condiciones) pero reconoce que es una idea muy interesante e intentarán llevarla a cabo en el futuro[4].

[4] http://www.elmundo.es/madrid/2015/09/10/55f1a53046163ff35a8b458a.html

1.3.4 Financiación

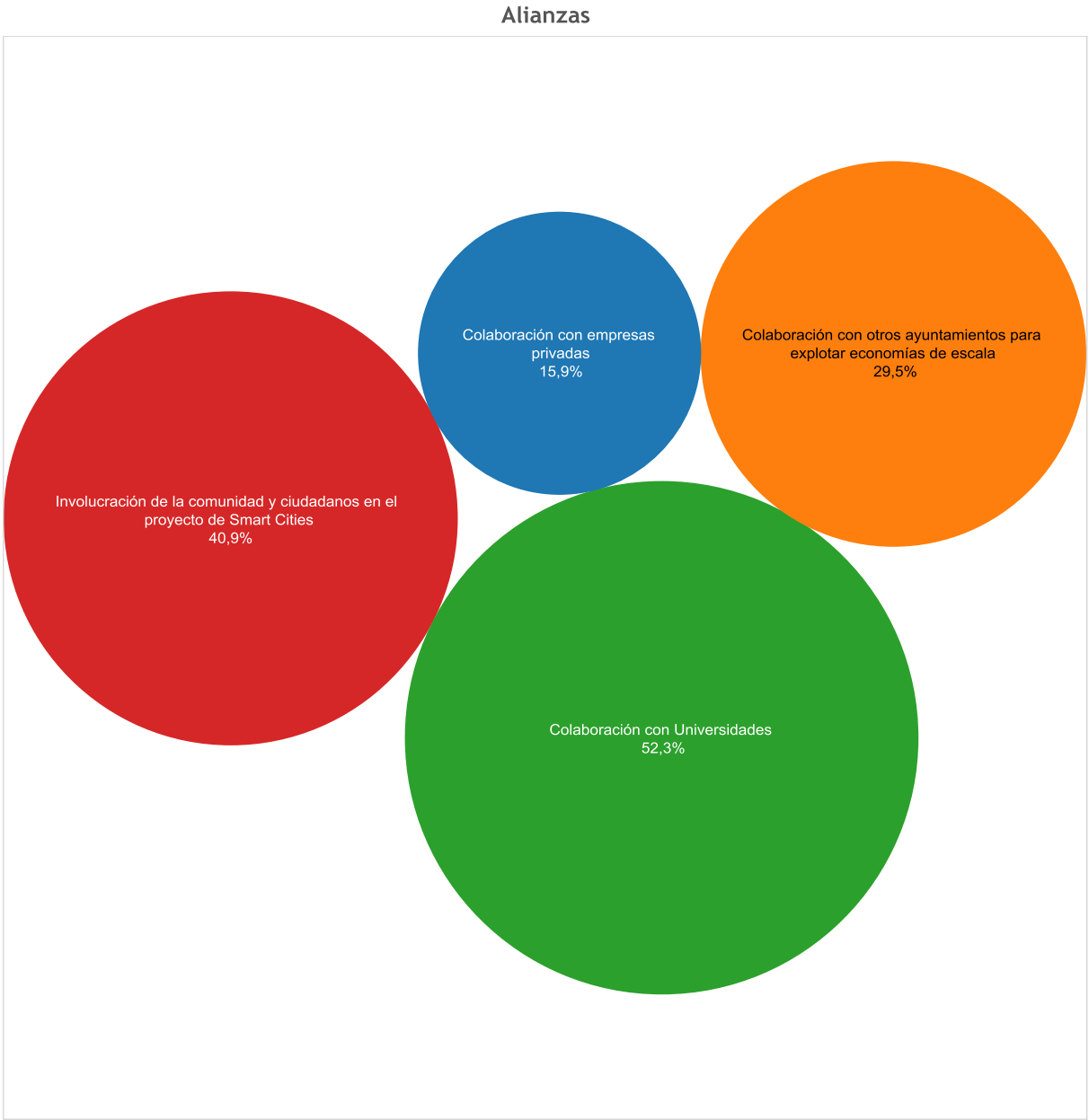
A continuación veremos datos relativos a la financiación, alianzas y recursos de las ciudades basado en un informe publicado en 2012 por IDC[5]:

- El reparto de las diferentes fuentes de financiación y el porcentaje del total de las grandes ciudades a las que llega cada ayuda:



[5] http://www.socinfo.es/contenido/seminarios/1404smartcities6/02-IDC_Smart_City_Analysis_Spain_2012.pdf

- Diferentes tipos de alianzas y porcentaje de grandes ciudades en las que se establecen:



- Recursos que se emplean para solucionarlas e ingresos que se pierden[6]:

  - En el año 2012, 2732 manifestaciones supusieron un gasto en seguridad de 1,9 millones de euros y en limpieza de 1,8 millones de euros.
  - En la manifestación del 25-S (2012) los cortes de tráfico afectaron a 60.000 vehículos y a unas 150.000 personas. Afectó además a 41 líneas de la EMT, un total de 476 coches y 89.165 viajeros. Se estima que la ciudad perdió por dicha manifestación más de 32.000 euros en recaudación ya que se estima la pérdida de 29.669 viajeros.
  - Una manifestación de 6.000 personas, según estimaciones de la Delegación de Gobierno, perjudica sólo en lo que se refiere a la EMT a 89.000 viajeros.
  - Las marchas del 22-M (2014) supusieron un coste de 166.000 euros y las cifras los gastos globales incluyendo seguridad y emergencias sanitarias (405.592 euros) y limpieza (89.655 euros) fueron de 655.000 euros.

[6] http://www.elconfidencial.com/espana/2012-09-27/madrid-10-manifestaciones-diarias-y-4-millones-de-gasto-en-limpieza-y-seguridad_218890/

2 MISIÓN, VISIÓN Y VALORES

2.1 Misión

Contribuir a mejorar la convivencia y bienestar ciudadano ayudando a la mejora de su movilidad.

2.2 Visión

Ser empresa de referencia para instituciones públicas, privadas y para los ciudadanos en el ámbito de las *Smart Cities*.

2.3 Valores

Innovación, compromiso social, acceso universal a las tecnologías, mejora continua.

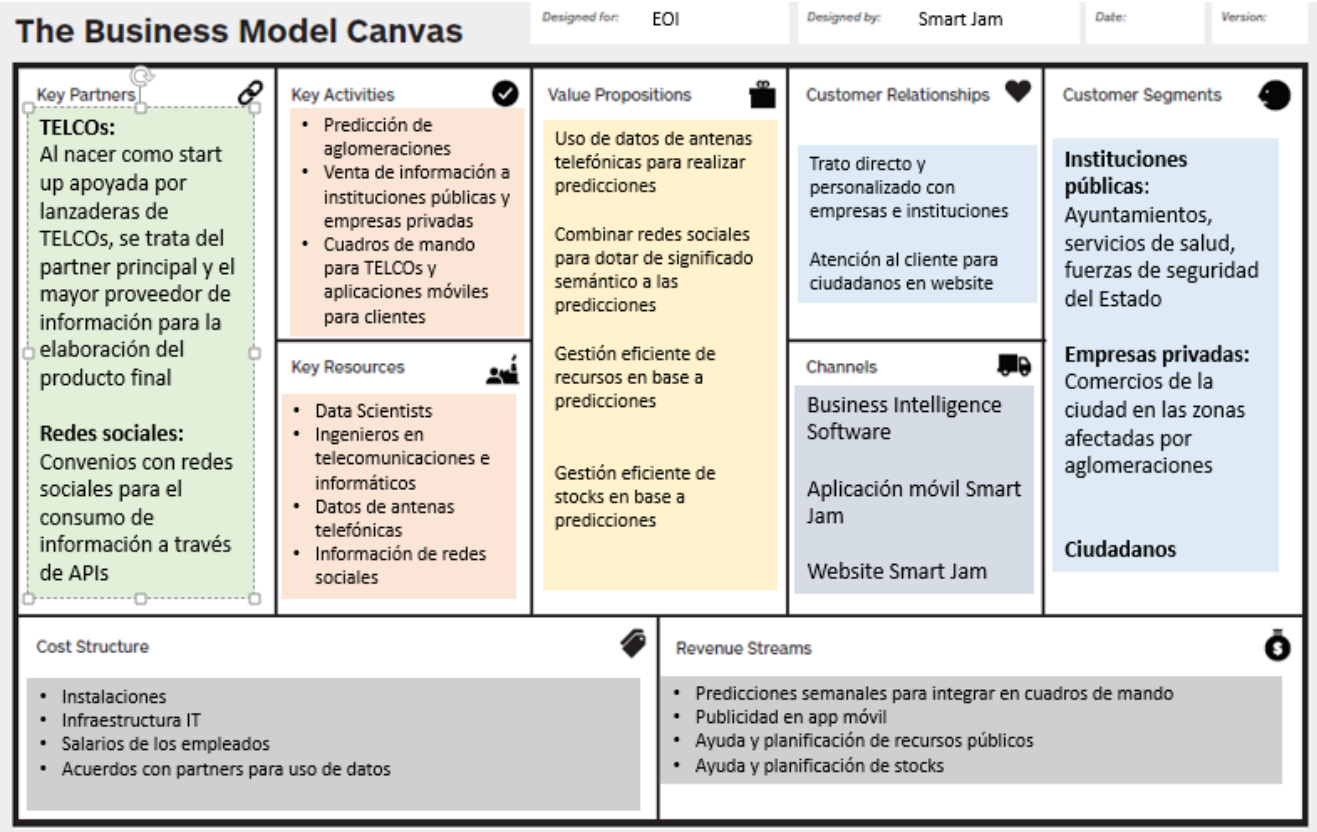
3 ANÁLISIS DAFO

Debilidades: • *Start Up* sin clientes afianzados. • Recursos financieros limitados. • Nuevos en el segmento de mercado. • Ausencia de alianzas empresariales.	**Amenazas:** • Mercado muy extendido de aplicaciones. • Veloz aparición de nuevas ideas desarrolladas. • Competencia de las grandes empresas de telecomunicaciones.
Fortalezas: • Uso innovador de la información de las redes sociales. • App orientada a la gestión eficiente de recursos. • Perfiles profesionales diferentes y complementarios entre los participantes. • Tecnologías empleadas de uso abierto y gratuítas en su mayoría.	**Oportunidades:** • Mercado incipiente de Smart Cities. • Posible evolución hacia empresa de servicios. • Resuelve un problema de carácter público. • Empresas propietarias de los datos necesitan el producto.

4 PLAN ESTRATÉGICO

4.1 Business Model Canvas

Antes de comentar en detalle el Plan Estratégico, es conveniente presentar el Business Model Canvas de nuestro modelo de negocio.



Como se puede observar, la alianza con las compañías telefónicas compone uno de los pilares de nuestro modelo de negocio para el desarrollo de las actividades principales que son aquellas que se basan en los datos facilitados por las mismas para la predicción de aglomeraciones, elaboración del consecuente cuadro de mando y venta a los diferentes clientes contemplados en el modelo de negocio. Los acuerdos con las redes sociales son de vital importancia para el consumo de la información georreferenciada que el propio ciudadano genera, dato muy enriquecedor para dotar de un significado a las aglomeraciones y tratar de explicar su naturaleza.

De esta forma, los clientes principales son las instituciones públicas y las empresas privadas que puedan sacar partido de las actividades que gestionan en su ámbito gracias a la ayuda de un cuadro de mando que les facilite la previsión, diagnóstico y organización de sus actividades de negocio. Las propias compañías telefónicas pueden resultar en proveedores de datos y consumidores del cuadro de mando, teniendo un doble rol en nuestro modelo de negocio. Por el contrario, los ciudadanos serían clientes de la aplicación móvil, aplicación que les ayuda a planificar su movilidad en la ciudad y anticiparse a posibles incidentes. El trato con empresas e instituciones públicas es personalizado debido a la propia naturaleza del producto que se oferta, un cuadro de mando personalizado según sus necesidades. Sin embargo, el trato con el ciudadano usuario de la App, es menos directo y personalizado, ya que es un producto estándar que se descarga y se consume. Al ser usuario de la App la idea es que actúe como proveedor de datos al facilitarnos su posición geográfica aceptando la política de descarga y uso de la app.

Los costes derivados de llevar a cabo la actividad de negocio son los habituales (instalaciones, salarios de empleados, infraestructura IT y pago por uso de la información), si bien es cierto que el perfil de los recursos humanos que se necesitan para desarrollar las predicciones, los Data Scientist, es un perfil difícil de encontrar, muy valorado y, por ende, tiene un coste más elevado de lo normal. De la misma forma, los datos procedentes de las compañías telefónicas son de difícil acceso y tienen un elevado coste y es por esto que se ha pensado en un modelo de negocio en el que las TELCOs no tengan un perfil de proveedor puro, sino más bien de partner actuando como una lanzadera de nuestra Start Up.

A continuación, se describen más detalladamente algunos de los puntos presentes en el Business Model Canvas y que conforman en Plan Estratégico de la Start Up.

4.2 ***Objetivos***

4.2.1 ***Descripción de la idea***

La idea de negocio consiste en el desarrollo de una herramienta para predecir en tiempo real dónde y cuándo se va a producir una concentración o aglomeración de personas y poder obtener la mayor información posible acerca de ella (tamaño, duración, motivo, etc.). Para lograrlo, se analizará la información procedente de varias fuentes de datos: redes sociales, instituciones públicas y datos procedentes de compañías de telecomunicaciones Esto permitirá gestionar de forma eficiente los recursos por parte de instituciones públicas. Así, podrán planificar las calles cortadas, los servicios de transporte afectados y también dedicar los recursos necesarios para un buen control de la aglomeración (accesos rápidos de policía, bomberos, servicios de salud, etc.) como ofrecer a los usuarios una aplicación para planificar cómo moverse por la ciudad evitando pasar por esos lugares de concentraciones. A su vez, se logrará evitar las circunstancias negativas que se suelen dar en ellas como pueden ser las pérdidas de tiempo, robos, incidentes, destrozos, etc.

Además, la información textual georreferenciada puede ser analizada, como en el caso de los *tweets* que incluyan coordenadas contando las palabras que más se repiten en un punto durante un tiempo determinado, para poder determinar la naturaleza u origen de la aglomeración. Esta información también puede cruzarse con los datos en tiempo real del tráfico e incidentes producidos en las vías ya que algunos ayuntamientos actualizan esta información con una frecuencia de unos cinco minutos. Para el funcionamiento de la aplicación será necesario almacenar los datos abiertos de tráfico e incidentes, los cuales crecerán exponencialmente a lo largo del tiempo. Es imprescindible almacenarlos para que los algoritmos de predicción puedan “aprender” de lo ya sucedido y así mejorar sus resultados en las siguientes predicciones. Para ello, se hará uso de tecnologías Big Data que nos permitan la gestión de grandes volúmenes de datos, procesos de minería de datos

y *Machine Learning* para la predicción de las aglomeraciones y para la extracción y análisis de la información georreferenciada de redes sociales. Todo ello intentando darles un uso innovador y un valor añadido.

Principalmente, consideramos que las partes interesadas en el negocio serían:

- **Los organismos públicos,** los cuales estarán interesados en la mejora de la seguridad de la ciudad, la reducción de gastos económicos, una mejor planificación de los transportes, disminución de los recursos empleados para el control de dichos eventos, etc.

- **Los ciudadanos** que pueden beneficiarse de la resolución del problema de las aglomeraciones evitando pasar por zonas donde se produzcan y buscando alternativas de transporte. Por otro lado, en ocasiones les puede interesar saber dónde y porque se producen estas concentraciones para poder acudir si son de su interés.

- **El sector privado** puede utilizar la información de las aglomeraciones y concentraciones para ofrecer sus productos de forma personalizada. Por un lado, conocerían el perfil de los manifestantes ya que se conocería el motivo de la concentración y, en segundo lugar pero no menos importante, el número de gente concentrada. Estos dos datos pueden ser de gran valor ya que les permitirá cuantificar y clasificar el público objetivo con lo que tendrán más fácil vender el producto adecuado.

Otro factor importante para el éxito de la idea y que hay que tener en cuenta es la existencia de una Red Española de Ciudades Inteligentes[7] que empezó a gestarse en 2011 con el objetivo de la mejora y desarrollo de las ciudades basada en la innovación y el conocimiento. Muestra de su progreso y crecimiento es que comenzó con 25 Ayuntamientos

[7] http://www.redciudadesinteligentes.es/

adheridos y en la actualidad son ya 62 los que forman parte de ella, incluyendo las principales ciudades. En su misión define las *Smart cities* como:

"Son Ciudades Inteligentes aquellas que disponen de un sistema de innovación y de trabajo en red para dotar a las ciudades de un modelo de mejora de la eficiencia económica y política permitiendo el desarrollo social, cultural y urbano. Como soporte de este crecimiento se realiza una apuesta por las industrias creativas y por la alta tecnología que permita ese crecimiento urbano basado en el impulso de las capacidades y de las redes articuladas todo ello a través de planes estratégicos participativos que permitan mejorar el sistema de innovación local."

Lo cual encaja perfectamente con el espíritu de nuestro proyecto, con lo que consideramos que se pueden establecer importantes colaboraciones con esta asociación de ciudades para el mutuo beneficio.

4.2.2 ***¿Qué ofrecemos?***

El negocio se concibe como una *Start Up* apoyada fuertemente en la tecnología Big Data para la gestión eficiente de recursos y servicios de una ciudad lo cual está fuertemente ligado al concepto de las *Smart Cities*.

La idea general es poder conocer cuándo, cómo, dónde y porque se está concentrando la gente para teniendo en cuenta aquello que se ve afectado (el estado del tráfico, incidentes que se estén produciendo, transporte, comentarios de los afectados, etc.) tomar decisiones. Esto puede ser de gran utilidad para los organismos públicos a la hora de gestionar los recursos y definir sus estrategias de actuación. Por otro lado, esta información puede ser usada también por el ciudadano cuando se encuentre a pie de calle para intentar anticiparse al problema o decidir qué hacer en ese mismo momento en función de sus planes, de las diferentes vías de escape o alternativas que hubiera ante un posible incidente, etc.

Más en detalle, la herramienta nos permite:

▫ Dar salida a grandes volúmenes de personas por diferentes vías de transporte, tanto público como privado, en base al análisis en tiempo real de cómo se concentran las personas.

▫ Prever seguridad y auxilio ante la previsión que nos proporciones de concentraciones de personas.

▫ Incentivar el consumo y el gasto ya que se ofrece información a los usuarios de eventos que están sucediendo en su ciudad, los sitios de los que está hablando la gente y por qué, actividades de ocio, etc.

▫ Tener un histórico de los puntos donde habitualmente existen concentraciones de usuarios, conociendo los hábitos y pudiendo adelantarnos a incidentes, situando en esos puntos, los medios adecuados para solventar los problemas que se hayan detectados.

▫ Organizar eventos relacionados y cercanos para que los ciudadanos puedan hacer rutas de actividades similares de forma sencilla, económica y segura.

Una vez el negocio se haya afianzado y haya evolucionado hacia un modelo más consolidado y estable, podría innovarse en la gestión de los recursos no orientados a servicios, como los energéticos (encendido y apagado de luces en función de la concurrencia de ciudadanos, apertura de semáforos en función del tráfico y pasos de cebra inteligentes que detectan cuando un ciudadano está cruzando o quiere cruzar para abrir o cerrar semáforos).

El ámbito de uso de la herramienta final comprenderá tanto su uso desde un puesto fijo como su uso desde dispositivos móviles. Desde un puesto fijo puede ser usado para planificación, control y organización de la ciudad por parte de las instituciones u organismos encargados de ello pero también el propio ciudadano puede usarlo para planificar sus acciones teniendo en cuenta la información que reciba. Otra vertiente de uso

sería desde los dispositivos móviles, en este caso se usaría para en tiempo real y con la información recibida, cambiar de planes o de trayecto, por ejemplo.

La aplicación estará basada en web para su uso en todo tipo de dispositivos. Para los dispositivos móviles (*Tablets* y *SmartPhones*) se desarrollará en una primera fase para el sistema operativo Android y luego para iOs. Para los PC's y servidores, de forma similar, se tendrá en primer lugar una versión para Windows y posteriormente para IO's y Linux.

Por tipo de uso, podemos clasificarlos en:

- *Smartphones:* Tendrá menos potencia y capacidad de visualización pero en cambio la gran ventaja de ser un dispositivo que siempre se lleva encima con lo que la comunicación de la información es instantánea.

- *Tablets*: aun siendo de reducido tamaño no es tan portable como los *Smartphone* pero lo compensa con la posibilidad de ver los cuadros de mando que muestre la aplicación en un tamaño más adecuado e interactuar con ellos de forma muy cómoda.

Servidores y PC's: En este caso disponemos de mucha más capacidad de visualización y potencia de cálculo pero no movilidad. Sería el más adecuado para centros de control y vigilancia de los organismos públicos que tendrían una visión global de la situación.

4.2.3 *¿Quiénes son nuestros potenciales clientes?*

Potenciales clientes

Nuestros potenciales clientes son:

▫ Usuarios de la aplicación, tanto ciudadanos como turistas, a los cuales les permite conocer qué está pasando en su ciudad (y qué es previsible que pase). Les ofrece información no sólo de donde se están produciendo las concentraciones sino también el motivo por el que se hayan producido. Esto lo consigue con el análisis semántico de la información de las redes sociales y ofreciendo información de qué se está hablando en esas zonas donde se concentra la gente. Por ejemplo, podría ser de gran interés para turistas para evitar zonas conflictivas, saber cómo se concentran o mueven personas de la misma procedencia y gustos, qué actividades se realizan ese día en la ciudad, etc. Las agencias de viajes serían a su vez, por tanto, potenciales clientes de la aplicación, ya que son los encargados de transmitir la información de la existencia de la aplicación a los clientes extranjeros (podremos ofrecerles rutas alternativas para visitar la ciudad sin tener que pasar por dichos puntos). El usuario final de la aplicación cuando se la descarga, nos da

permisos mediante la aceptación de la política de la misma a registrar la posición de su dispositivo cada período de tiempo para saber a su vez cómo se mueve, saber desde dónde manda una alerta en caso de que quiera hacerlo y juntar estos datos de posición junto con la información georreferenciada extraída de las redes sociales.

▫ Instituciones públicas como el Ayuntamiento de una localidad, la Policía, el cuerpo de Bomberos, etc. serían a su vez clientes, ya no solo de la aplicación, sino de la propia empresa contratando sus servicios. Aparte de poder hacer un uso de la aplicación como cualquier otro usuario (ciudadano o turista) tendría la posibilidad de adquirir un cuadro de mando personalizado desde donde tener acceso y control a toda la información de interés para la planificación y gestión del correcto funcionamiento de la ciudad.

▫ Empresas: sin duda se les presentaría la oportunidad de vender sus productos de forma personalizada y directa a un sector de la población muy localizado en tiempo y lugar. Tengamos en cuenta que dispondrían de mucha información interesante proveniente de las concentraciones como: donde se producen, que cantidad de gente y que perfil tiene, el motivo de la concentración, etc. Con todo ello podría llevar a cabo campañas tanto de venta de productos o servicios *in situ* como campañas de publicidad de productos o eventos a los que asistir posteriormente. De un modo más clásico también podrían suponer una fuente de ingresos por la venta de espacios publicitarios en la propia aplicación.

4.2.4 ¿Cuál es nuestra ventaja competitiva?

La propia orientación del producto hacia el concepto de *Smart Cities* es una ventaja competitiva en sí, ya que es un modelo que aún no tiene demasiada aplicación en España a nivel de gestión inteligente de los servicios públicos. Sí que existen iniciativas y pilotos en algunas ciudades, como se ha comentado anteriormente, existen unos 60 ayuntamientos

adheridos a la Red Española de Ciudades Inteligentes en continuo crecimiento, lo que da una idea del recorrido del negocio.

Los diferentes perfiles profesionales que forman parte del proyecto proporcionan una gran amplitud y diversidad de conocimientos que van desde la topografía, herramientas de visualización, marketing y gestión de equipos hasta los más los técnicos dentro de los campos de la informática y telecomunicaciones. Esto permite, aun siendo un equipo reducido, contar con especialistas en los principales campos que abarca el proyecto y, a su vez, dota al equipo de una gran capacidad de adaptación e innovación.

La apuesta por fuentes de datos abiertas (*Open Data*) es otra ventaja competitiva al ser muy reciente su aparición y preverse un fuerte crecimiento de su uso por parte de las instituciones tanto públicas como privadas posicionándonos en un buen lugar para su futuro aprovechamiento. El uso de tecnologías de desarrollo abiertas (*Open Source*) nos proporciona también la capacidad de adaptación y flexibilidad que permitirá una sencilla integración con otras tecnologías y fuentes de datos públicas con un coste reducido.

4.2.5 ***¿Cuál es nuestro ámbito geográfico de acción?***

Al concebirse la idea de negocio como una *Start Up*, consideramos adecuado el desarrollo y aplicación de la misma en un ámbito reducido y viable, tratando de no ser demasiados ambiciosos; es preferible identificar un área geográfica viable y con la facilidad de obtención de datos para la toma de decisiones basadas en los mismos.

Para la creación y los primeros pasos del desarrollo de la herramienta hemos usado los datos de una ciudad grande y capital de un país como Roma. Gracias a la participación en el concurso *TIM Big Data Challenge 2015*[8] hemos tenido acceso a fuentes de datos acerca de las comunicaciones y movilidad de los ciudadanos lo cual nos ha resultado de gran ayuda a la hora del desarrollo de nuestra aplicación.

[8] http://www.telecomitalia.com/tit/en/bigdatachallenge/news-social/big-data-jam.html

Aparte de este camino ya abierto consideramos que la ciudad de Madrid se plantea como una excelente opción por los siguientes motivos:

- Su Ayuntamiento está adherido a la Red Española de Ciudades Inteligentes debido a su interés en este campo.

- Tiene en marcha alguna iniciativa ya orientada a las *Smart Cities* como la instalación de sensores que miden la contaminación para así poder anticiparse a altos niveles mediante la toma de decisiones estratégicas (la entrada gratuita al transporte público ciertos días, prohibir la entrada de vehículos al centro, etc.).

- Se está apostando por compartir una gran cantidad de datos *Open Data* (muchos de ellos en tiempo cuasi real).

- Tiene una extensión y censo suficiente como para que se genere una cantidad considerable de información georreferenciada.

- Tiene distritos y barrios muy heterogéneos lo cual facilita la obtención de información de interés en muchos de ellos. Una posibilidad interesante sería comenzar por un distrito o barrio en lugar de toda la ciudad de Madrid para así disponer de un volumen de datos más fácilmente tratable. Por otro lado, poder elegir, de entre los diferentes barrios y distritos, los más representativos e interesantes para nuestros objetivos puede ser de gran ayuda para mejorar la eficiencia de nuestra aplicación.

- Cuenta con un gran atractivo turístico por ser la capital de España.

Una vez la herramienta se haya desarrollado para ciudades como Roma o Madrid su adaptación a otras grandes ciudades será relativamente sencilla. En una segunda fase se afrontaría su implantación a ciudades de menor tamaño y que probablemente requieran una mayor precisión por parte de la herramienta debido al menor volumen y calidad de los datos que en ellas se generan.

5 PLAN DE ACCIÓN

5.1 Alcance

Obtención de un cuadro de mando con la herramienta Qlik Sense que implemente los análisis predictivos obtenidos a través de la explotación de la información de antenas telefónicas en la ciudad de Roma, la contabilización de palabras procedentes de tweets geolocalizados, los negocios de la ciudad de Roma y los puntos de interés.

5.2 Análisis del talento humano

Para el desarrollo del proyecto se cuenta con los cuatro integrantes que conforman el equipo de trabajo fin de máster: Jesús Villagra (ingeniero técnico informático), Ana Cantero (ingeniero técnico en topografía), Ignacio Bonet (ingeniero informático) y Javier Lamana (ingeniero técnico informático).

5.3 Análisis de los recursos físicos

5.3.1 Arquitectura

5.3.1.1 Infraestructura

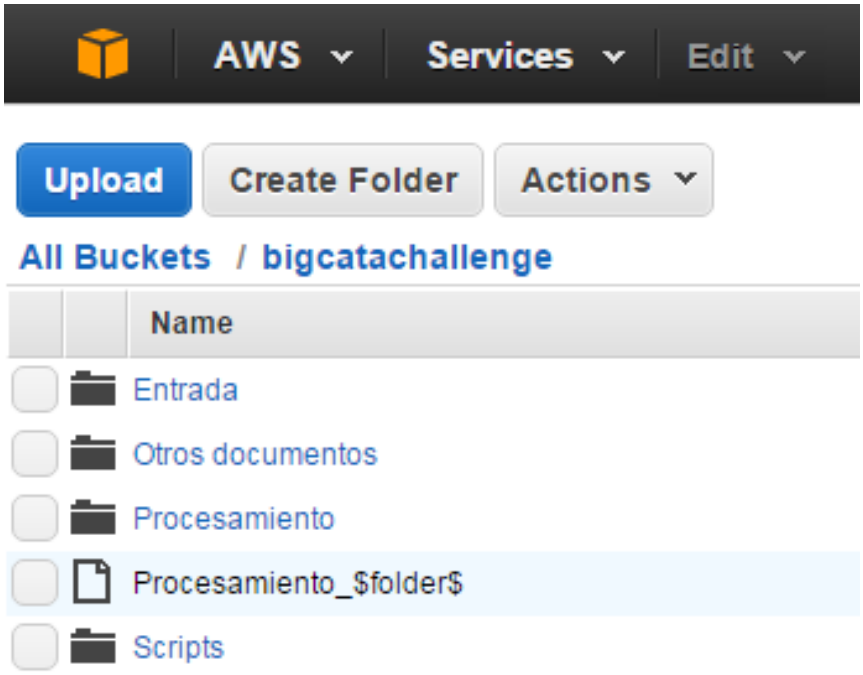
La arquitectura empleada para el proyecto se ha implementado en su totalidad en *Amazon Web Services*, gracias a la flexibilidad, escalabilidad y facilidad de desarrollo e implementación que ofrece la nube, así como el mantenimiento de toda la parte física de Hardware por parte de Amazon y la total abstracción de la parte más compleja de configuración de sistemas.

5.3.1.2 Servicios empleados

S3

Con el fin de almacenar y consumir todo el volumen de datos generado durante la fase del proyecto, se ha empleado el sistema de ficheros distribuido de Amazon, que permite un

almacenaje escalable, seguro y económico, que puede ser accedido en función de los permisos que se les otorgue a los usuarios definidos. Dentro de S3 pueden definirse diferentes buckets con diferentes políticas de acceso y comunicación, así como usuarios. La estructura de ficheros empleada es la siguiente:



Estructura de ficheros en S3. 1

Dentro del directorio 'Entrada' se encuentran todas las fuentes de datos en origen sin tratar. El directorio 'Procesamiento' contiene todos los outputs generados en las diferentes etapas del proyecto. 'Otros documentos' y 'Scripts' tienen el fin de almacenar documentos del proyecto empleados para desarrollar el mismo.

EC2:

Las instancias EC2 son muy útiles para muchos fines, pero en el caso que nos ocupa nos ofrecen una alta capacidad de computación y una memoria RAM adecuada para correr procesos y scripts que de otra forma consumirían mucho tiempo.

Así, se han empleado máquinas EC2 para la descarga de ficheros de la plataforma del concurso[9], su descompresión y su correspondiente subida a S3, para correr scripts de Python que necesitan trabajar con ficheros muy grandes (parseo de json a formato tabla para dejar lista la entrada de la ETL) y, por último, como instancias componentes de un cluster para ejecutar trabajos en paralelo de EMR (*Amazon Elastic Map Reduce)* para ejecución de ETL.

Así, se han empleado máquinas EC2 para:

- Descarga de ficheros de la plataforma del concurso, su descompresión y su correspondiente subida a S3.
- Ejecutar scripts de Python que necesitan trabajar con ficheros muy grandes (*parseo* de json[10] a formato tabla para dejar lista la entrada de la ETL).
- Como instancias componentes de un clúster para ejecutar trabajos en paralelo de EMR (*Amazon Elastic Map Reduce)* para ejecución de ETL.

Las máquinas empleadas en la mayoría de los casos, excepto en *EMR*, son del tipo m4.xlarge.

[9] https://dandelion.eu/
[10] JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos.

Las instancias M4 son la última generación de instancias de uso general. Esta familia proporciona un equilibrio de recursos informáticos, memoria y redes. Características:

- Procesadores Intel Xeon® E5-2676 v3 (Haswell) de 2,4 GHz.
- Optimizados para EBS por defecto sin coste adicional.
- Soporte para redes mejoradas.
- Equilibrio entre recursos de informática, memoria y red.

Modelo	vCPU	Memoria (GiB)	Almacenamiento en SSD (GB)	Rendimiento EBS dedicado (Mbps)
m4.xlarge	4	16	Solo para EBS	750

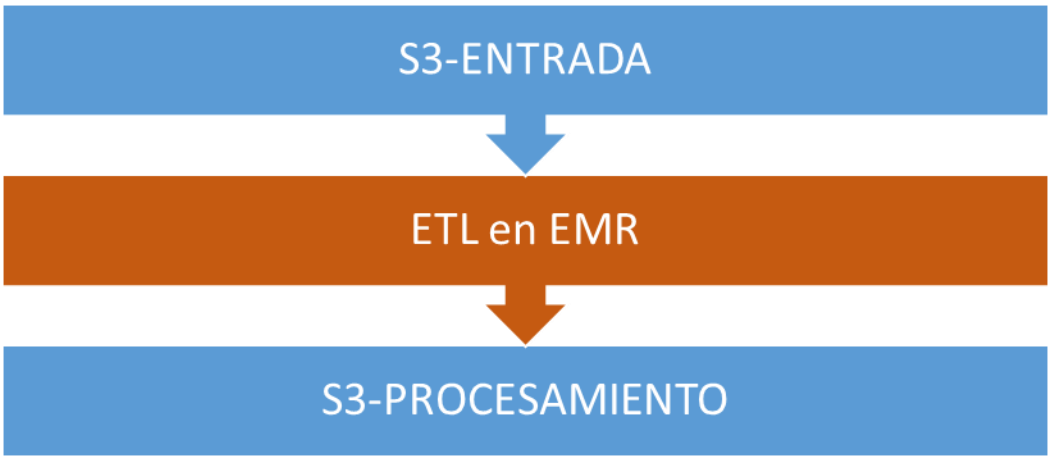
EMR:

Amazon Elastic MapReduce (Amazon EMR) es un servicio web que facilita el procesamiento rápido y rentable de grandes cantidades de datos. En realidad, se trata de una distribución gestionada de Hadoop en la que toda la instalación y configuración han sido realizadas. Además, tiene la opción de instalar Hive, un lenguaje parecido a SQL que permite simular este tipo de consultas sobre grandes cantidades de datos en paralelo evitando tener que desarrollar los trabajos de Map/Reduce en Java propios de Hadoop, ya que el propio lenguaje se encarga de traducir la consulta SQL a Map/Reduce.

EMR es de gran utilidad ya que permite que el proceso de ETL sobre grandes cantidades de datos, es decir, el procesamiento masivo de ficheros, sea ágil y mucho más rápido que con cualquier herramienta convencional no preparada para soportar grandes volúmenes de información.

La nube nos permite configurar en tan solo unos minutos un cluster de x nodos que se encargará de procesar en paralelo las consultas, procesos y transformaciones que se desee sobre los ficheros que se encuentren en el sistema de ficheros distribuido propio de Hadoop (HDFS) o sobre otro sistema de ficheros, como en nuestro caso, en S3.

El flujo de trabajo sería el siguiente:



Workflow. 2

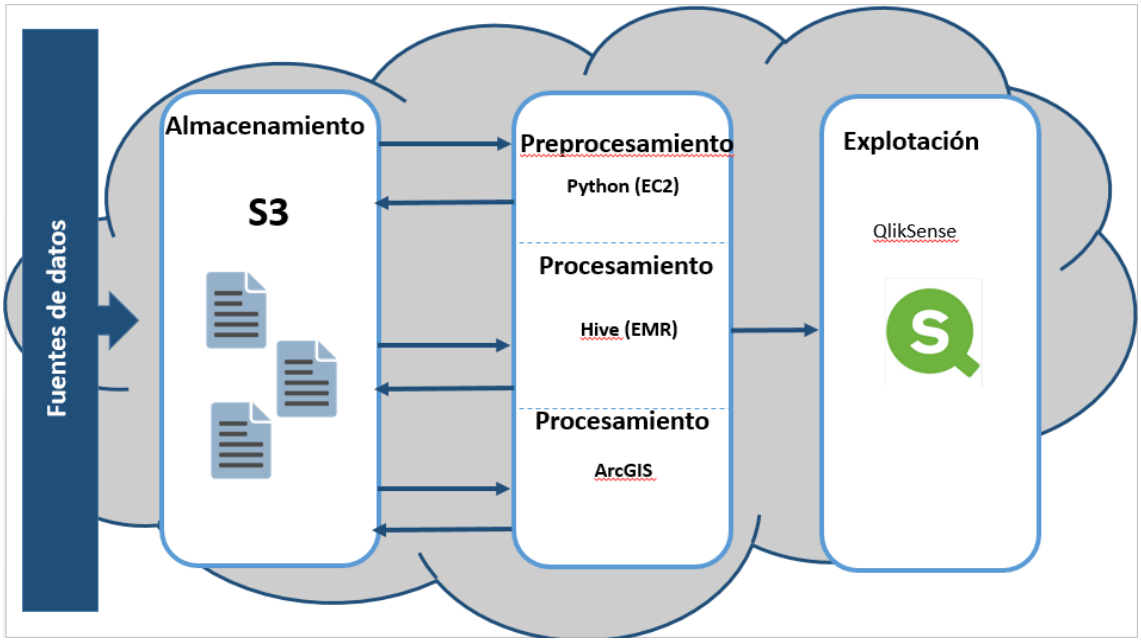
REDSHIFT:

La explotación de los datos, es decir, la fase de análisis, descubrimiento de información y generación de conocimiento mediante el consumo del dato, debería de realizarse en un gestor de base de datos, de forma que fuese posible consumir la información desde una herramienta de BI, por ejemplo. Cuando existe una gran volumetría de datos, muchos orígenes diferentes y muchas relaciones entre los mismos, en definitiva, existe un modelo de datos lógico complejo, debería implementarse un modelo físico sobre una base de datos que facilitase su explotación. Sin embargo, para el desarrollo del pequeño prototipo que en este proyecto se presenta, no existe tal complejidad para que haya que recurrir a este tipo de herramienta para consumir la información (ya que la propia herramienta de BI empleada permite la carga del fichero en bruto sin tener que conectarse una base de datos). Aun así, creemos necesario recalcar que en un futuro y para un proyecto real, sería necesario en el ámbito del proyecto definir un gestor de base de datos adecuados.

Una de las opciones que ofrece AWS es Redshift, un almacén de datos rápido y totalmente gestionado a escala de petabytes.

Amazon Redshift ofrece un rendimiento de consulta rápido gracias a la utilización de la tecnología de almacenamiento en columnas para mejorar la eficacia de E/S y realizar consultas en paralelo entre varios nodos. Dispone de controladores JDBC y ODBC. La velocidad de carga de los datos aumenta de manera lineal con respecto al tamaño del clúster, con integraciones en *Amazon S3* y *Amazon Elastic MapReduce*.

5.3.1.3 Estructura



Arquitectura en la nube de AWS. 1

5.4 Actividades y Tareas

5.4.1 Recopilación de fuentes de datos

5.4.1.1 Orígenes de datos

Para el desarrollo del presente proyecto se ha tenido acceso a diferentes fuentes de datos, tanto información pública obtenida a través de *web scraping*[11], como información privada mediante el acceso a la plataforma de acceso restringido del concurso al que se presenta la idea.

El ámbito del proyecto se centra en la ciudad de Roma ya que para la obtención del Mínimo Producto Viable (MVP), en este caso un cuadro de mando, es más que suficiente al enfocarse en una ciudad con una gran población, rica en turistas, con una gran riqueza y patrimonio cultural, una gran oferta comercial, lo suficientemente grande como para generar un gran volumen de información y, asimismo, que el público objetivo de dicho producto sea lo bastante amplio como para introducir el producto en el mercado.

5.4.1.2 Web Scraping

Una de las formas que hemos considerado para la obtención de fuentes de datos ha sido mediante la búsqueda en páginas web de eventos, actividades culturales y cualquier otra en la que encontráramos no solo la información de los datos buscados, sino también las coordenadas geográficas del lugar donde se sitúa el dato, esto es necesario para poder ubicar la información dentro de nuestro mapa de Roma. Para todo ello hemos desarrollado un algoritmo en *Python* usando la librería "urllib" para acceder a las diferentes webs en las que se encuentra la información a extraer y generar el fichero con formato *json* con el que vamos a trabajar.

[11] **Proceso de recopilar información de forma automática de la Web.**

5.4.1.3 Web Semántica

Otro medio para obtener información, en este caso de puntos de interés como parques, monumentos, jardines, museos, zonas deportivas, etc., lo encontramos dentro de la página Web[12], la cual nos permite usando el lenguaje de consulta *SPARQL (Protocol and RDF Query Language)* obtener los datos anteriormente descritos. La sentencia usada acota la consulta a la ciudad de Roma, y nos ofrece la información del tipo de dato: jardín, monumento, restaurante, etc. La dirección postal, y la longitud y latitud, que nos da la oportunidad de situarla en el mapa de Roma.

Adjuntamos la sentencia utilizada:

```
SELECT DISTINCT * WHERE {
  ?s dbpedia-owl:location dbpedia:Rome;
   rdf:type ?tp;
    dbpedia-owl:address ?address;
dc:title ?title;
geo:lat ?lat;
geo:long ?long.
}
```

5.4.1.4 Plataforma del concurso Big Data Challenge 2015 (Telecom Italia)

Telecom Italia pone a disposición de los participantes del concurso una serie de datasets de diversa índole y temática que hacen referencia a diferentes ciudades y ámbitos geográficos de Italia. El fin es que (haciendo uso de al menos un *dataset*) se obtenga un uso competitivo, innovador y pionero de los datos, de forma que se le dé un valor añadido a la información que se pone a disposición de los usuarios.

[12] http://tour-pedia.org/about/lod.html

De cara a lograr el objetivo final del proyecto, y siempre siguiendo la línea de la misión definida que no es otra que mejorar la convivencia y bienestar ciudadano resolviendo los problemas ocasionados por la aglomeraciones no controladas, se estudian los recursos disponibles en la plataforma *Dandelion*[13]:

Tras el estudio de cada uno de los conjuntos de datos disponibles sobre la ciudad de Roma, se decidió utilizar aquellos que ayudasen de alguna forma a definir el número de personas que se mueven por una ciudad y cómo se mueven, siendo fundamental la línea temporal para poder extraer algún tipo de conclusión y orientar las acciones de entidades públicas y privadas con el fin de que se pueda llegar a la toma de decisiones basada en datos de una forma inteligente y eficaz.

De esta forma, los datasets empleados son los siguientes:

Grid de Roma: Se trata de un archivo en formato shapefile (formato nativo del software de Sistemas de Información Geográfica (SIG) de ESRI) que contiene los polígonos que conforman la malla de telecomunicaciones que puebla la ciudad de Roma. Cada cuadrícula de la rejilla lleva asociado un ID de cuadrícula. La superficie de la rejilla no es uniforme, ya que en las zonas más pobladas y donde se registra un mayor número de señales las cuadrículas son más pequeñas, y aquellas zonas menos pobladas y donde se registra una menor actividad, tienen una mayor superficie.

Presencia: El archivo de presencia contiene el cómputo de presencia en base a llamadas, mensajes y conexiones a internet cada 15 minutos en cada cuadrícula del GRID. Si una señal idéntica se registra en dos cuadrículas, es decir, la persona se mueve de una cuadrícula a otra, su presencia se contabiliza en ambas cuadrículas. Originariamente, el

[13] https://dandelion.eu/

campo que indica la fecha viene en formato UNIX en milisegundos (número de milisegundos desde el uno de enero de 1970). Además, el campo presencia venía dividido entre 10.

Este *dataset* será el que compondrá el eje del proyecto y el que determinará cómo se aglomeran las personas dentro de la ciudad, en qué horarios y circunstancias y servirá para dar apoyo a particulares, empresas y entidades públicas.

Negocios: El fichero csv de negocios contiene información de todos los negocios de la ciudad de Roma, así como de cada una de sus ubicaciones en caso de ser un negocio que cuente con más de una ubicación. Este *dataset* se utilizará a nivel de cuadro de mando para representar las entidades y ser capaz de ofrecer un servicio a aquellas empresas para las que se prevea una aglomeración o un cúmulo de personas poco frecuente en la cuadrícula del GRID en la que se ubica dicho negocio.

Tweets: La plataforma también pone a disposición de los usuarios una API para acceder a la información de los tweets de diferente forma. En un principio, el objetivo del proyecto, que es la predicción de aglomeraciones y concentraciones no controladas, iba a obtenerse a partir de la obtención, tratamiento y análisis de los tweets geolocalizados en la ciudad de Roma. Sin embargo, tras estudiar las posibilidades de obtener un volumen adecuado como para poder llegar a extraer conclusiones robustas a raíz de dicha información, se vio que el volumen de tweets geolocalizados no era suficiente. Además, existían posibilidades mucho más atractivas para llegar a conclusiones afianzadas y de peso sobre cómo se concentra la población y dónde en un momento determinado, como es la información de la presencia descrita anteriormente. El sesgo en este *dataset* es mucho menor ya que, si bien es verdad que no todo el mundo posee un dispositivo móvil, es mucho menor la cantidad de personas que generan información geolocalizada en las redes sociales.

No obstante, Twitter, entre otros, es de gran ayuda para dotar de un significado semántico a las conclusiones extraídas mediante el estudio de otros *datasets*, ya que proporciona texto emitido por los propios usuarios de la ciudad.

Así, mediante uno de los métodos disponibles en la API para extraer información de esta red social, se extrajeron todos los tweets alrededor de las coordenadas de un punto central de Roma en 100 kilómetros a la redonda.

El resultado fueron unos 100.000 tweets geolocalizados en los meses en los que se centra el ámbito de desarrollo del proyecto (febrero, marzo y abril de 2014). La información devuelta por la API es en formato JSON y no proporciona el contenido del tweet en sí, sino recursos de la *DBpedia*[14] que casan con las palabras contenidas en dicho texto. Así, la fase homogeneización para el conteo de palabras para cada cuadrícula del GRID para cada hora, es mucho más sencilla.

5.4.1.5 Estructura de los datos (antes de ETL)

GRID ROMA (shapefile)		
Id_grid	string	ID de la cuadrícula del grid para el que se calcula la presencia
Geometría	geom	Geometría del polígono que define cada cuadrícula del GIRD

Datos desde: 2015-02-28 23:00:00 hasta: 2015-04-30 21:45:00

PRESENCIA (csv)		
Id_grid	string	ID de la cuadrícula del grid para el que se calcula la presencia
double_presencia	double	Presencia asociada a ID del GRID. Dividida entre 10
date_unix_fecha	date	Fecha en milisegundos

[14] http://es.dbpedia.org/

		UNIX

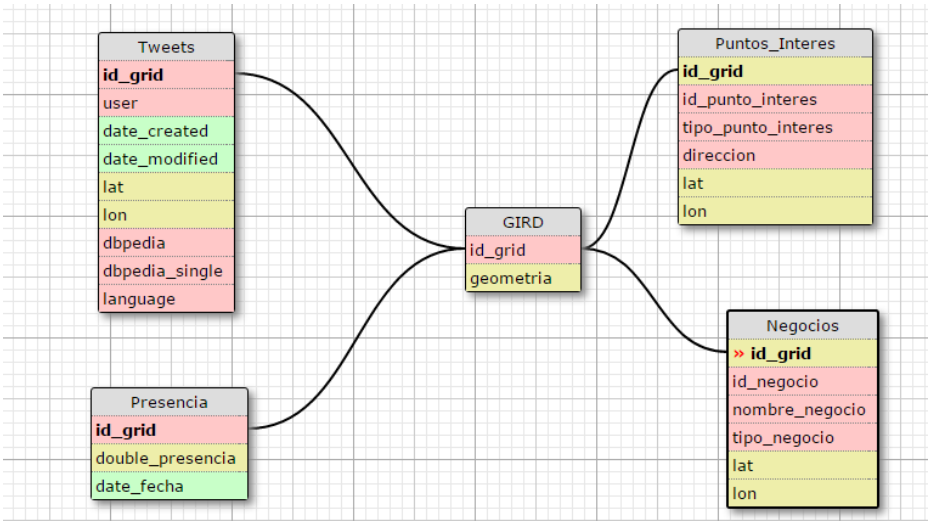
Negocios (csv)		
subject_id	string	ID de la empresa
sign	double	Nombre de la empresa
long	double	Coordenada longitud donde se ubica la sede
lat	double	Coordenada latitud donde se ubica la sede
kind	string	Área de negocio de la empresa
area	string	Campo prescindible

Tweets (json)		
created	date	Fecha de emisión del tweet
entities	string	Recursos de la DBpedia asociados al tweet
geometry	float	Coordenadas del tweet
language	String	Idioma del tweet
modified	date	Última fecha de modificación del tweet
timestamp	date	Fecha en formato UNIX de emisión del tweet
user	string	ID que identifica al usuario emisor del tweet

Puntos de Interés (json)		
ID	string	Identificador del punto de interés
tp_value	string	Tipo de punto de interés
Address_value	string	Dirección del punto de interés
title	string	Nombre del punto de interés
Coordx	double	Coordenada longitud del tipo de interés
coordy	double	Coordenada latitud del punto de interés

5.4.1.6 Modelo de datos

El modelo de datos lógico considerando los atributos necesarios de cada entidad queda plasmado de la siguiente manera:



Modelo de datos. 1

Se trata de un modelo muy sencillo, fácil de entender y de implementar, suficiente para el desarrollo de un primer prototipo del producto que sea capaz de plasmar la idea e introducirla en el mercado objetivo.

5.4.2 Tratamiento de las fuentes de datos

Para llegar al modelo lógico de datos definido, se han seguido una serie de etapas:

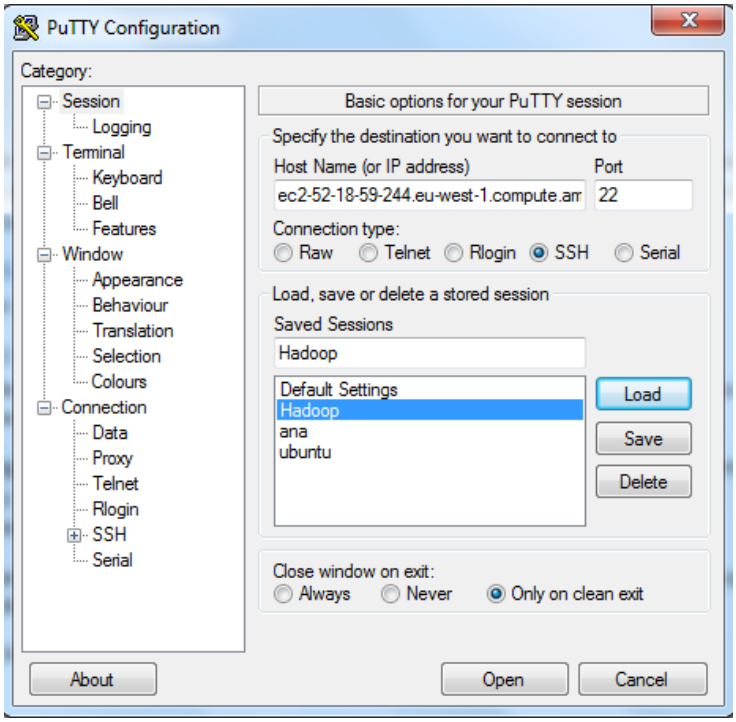
5.4.2.1 Carga de datos en sistema de ficheros distribuido de Amazon (S3)

Una vez decididas y definidas las fuentes de datos que iban a utilizarse, se procedió a la carga de los ficheros en S3.

Sin embargo, gran parte de estos ficheros procedían de datos facilitados por la plataforma del concurso y tenían un tamaño lo suficientemente grande como para que la descarga, descompresión y subida de ficheros a S3 fuera lenta y costosa.

Por ello, para esta primera etapa se utilizó el servicio de *Amazon Web Services EC2*, ya que la descarga de ficheros y su subida a S3 se realiza a una mayor velocidad.

Para ello, basta con levantar el servicio EC2, conectarse vía *Putty* (cliente SSH), y proporcionar la URL de descarga mediante el método Linux 'get'.



Conexión SSH vía *Putty*. 1

Una vez descargados los ficheros, se descomprimen y trasladan a S3 haciendo uso de la interfaz de comandos de Amazon que permite comunicar instancias EC2 con el sistema de ficheros S3 *(AWS CLI, http://docs.aws.amazon.com/cli/latest/)*.

5.4.2.2 Tratamiento de ficheros previo a ETL

Los archivos provenientes de llamadas a APIS, web *scraping* se encuentran en formato JSON. En ocasiones, cuando la estructura del JSON está muy anidada, se hace difícil consumir este tipo de archivos desde *Hive* (Data warehouse Software) con lo que es mejor realizar una tarea previa para trasformar los datos a una tabla. En otras ocasiones, cuando

la estructura no está muy anidada, *Hive* puede leer JSON siempre y cuando exista una sola línea por registro o entidad.

En el caso de los puntos de Interés, se partía de un BSON de *MongoDB* (JSON particular de *MongoDB*) con varias líneas por entidad. La estructura del BSON no era muy anidada y transformado cada punto de interés a una sola línea, *Hive* es capaz de leer e interpretar los campos del BSON y transformarlos internamente a una tabla para trabajar con ellos.

En el caso de los tweets, con una estructura más compleja, fue necesario transformar el JSON a una tabla de n filas y m columnas con los atributos definidos.

Ambos scripts se ejecutaron en Python en una máquina EC2 que agilizase el proceso.

Dichos scripts se pueden encontrar en los anexos:

- *Parse Tweets* (ver anexo)
- *Parse Puntos Interes* (ver anexo)

La salida de ambos procesos será una entrada, ya que no se ha transformado el dato sino adecuado a los requerimientos de procesamiento de la herramienta de ETL.

5.4.2.3 ETL

El proceso ETL consta de dos fases diferenciadas:

- Procesado, limpieza y transformación de atributos (proceso en EMR con *Hive*)
- Asociación, con la ayuda de un software de GIS, de un campo ID_GRID a aquellas entidades que no posean en origen dicha información (proceso con ArcGIS, de ESRI)

5.4.2.3.1 ETL en *Hive*

El proceso ETL que procesa, limpia y transforma los campos de las entidades que

componen el modelo de datos se desarrolla en un cluster de EMR (Elastic Map Reduce de Amazon Web Services). Para facilitar la paralelización de las tareas, se recurre a un lenguaje parecido a SQL denominado Hive que permite realizar las consultas y transformaciones pertinentes en un lenguaje mucho más sencillo que los jobs de Map/Reduce de Hadoop.

Además, en el caso de los t*weets Hive* ha sido utilizado para procesarlos y contar cuánto se repite cada palabra por cada id de grid y hora de cada día durante los meses para los que se tienen datos.

En total, se generaron tres procesos de ETL:

-ETL de las entidades 'Presencia' y 'Negocio' (ver anexo Fuentes_ETL)

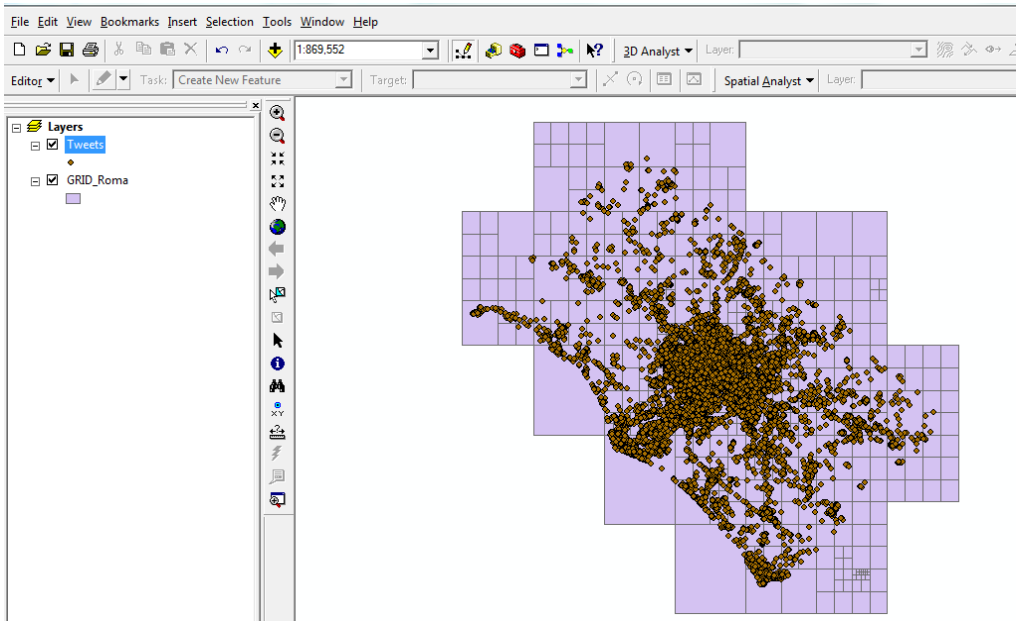
-ETL de las entidades que necesitaron de un parseado previo en Python, 'Eventos' y 'Puntos de Interés'. La entidad 'Eventos' no forma parte del modelo finalmente ya que no se consiguió generar un número suficiente de los mismos mediante las técnicas de web *scraping* que trataban de obtener dicha información de diferentes portales de interés (ver anexo JSON_ETL)

-ETL para realizar un procesado de los tweets y contar palabras por grid y hora (ver anexo *Tweets_ETL_WordCount*). Esta ETL es NECESARIO realizarla a la inversa, es decir, en primer lugar con la salida generada en Python se carga el fichero en el software de GIS, se calcula un ID_GRID para cada registro y se procede a realizar el conteo de palabras en EMR.

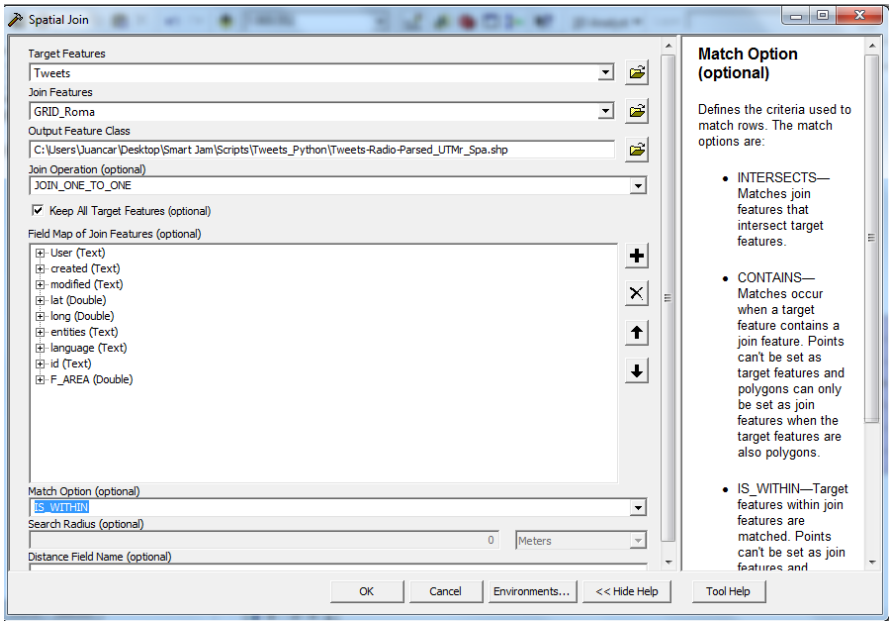
5.4.2.3.2 ETL en ArcGIS

Una vez generadas las salidas en proceso de ETL anterior, aquellas entidades para las que no se tiene un id de grid, es decir, se tienen únicamente sus coordenadas geográficas pero no dentro de qué grid cae el registro, se tratan con una herramienta de GIS para realizar un *join* espacial entre la entidad geométrica del Grid y la entidad a tratar.

Esta operación ha sido necesaria realizarla para las entidades ‘Tweets’, ‘Negocios’ y ‘Puntos_Interes’



ETL en ArcGIS. 1



Join espacial entidades. 1

5.4.3 Proceso Predictivo

Una vez generadas las salidas producidas al lanzar las diferentes ETL´s, comenzamos con la exploración y el estudio de los datos.

Fases en la que nos hemos basado para el análisis predictivo en el estándar (CRISP-DM):

5.4.3.1 Comprensión del negocio

Esta parte está bastante explicada en apartados anteriores, se indica un breve resumen: nuestro negocio se basa en los datos de las conexiones telefónicas realizadas por individuos de la ciudad de Roma en un periodo determinado, y nuestro fin es procurar un sistema de decisión inteligente, con el fin de adelantarnos a posibles incidentes por la concentración de personas en un punto determinado.

5.4.3.2 Comprensión de los datos

Una vez obtenidos los *datasets* ajustados a nuestras necesidades, comenzamos el análisis básico de las variables, y de los datos. El *dataset* de presencia contiene la siguiente información:

Count	**5.156.900**
Mean	**1909.330225**
Std	**2271.824614**
Min	**25.082298**
25%	**839.621708**
50%	**1475.829206**
75%	**2326.811958**
Max	**58033.831608**

Podemos ver que tenemos más de 5 millones de registros, también vemos que el valor de presencia máximo es de 58033.831608, comprobamos en que grid de Roma ocurre ese valor tan alto de concentración de personas.

GRID: 3447_2 FECHA: 2015-04-30 21:45:00

Si estudiamos este grid, comprobamos que es más grande que el resto de los grid que tenemos, investigando sobre ello, comprobamos a que es debido a como está tomada la información y como están formadas las rejillas. La dimensión de las rejillas está creada en base al alcance de las antenas de telefonía, es decir, en zonas menos pobladas la amplitud del grid es mayor, ya que con una antena cubren todas los posibles accesos a ellas, en cambio en zonas muy pobladas, la dimensión del grid es menor.

Un vistazo de 2 filas del *dataset*:

GRID	FECHAHORA	PRESENCIA
03430_0_3	2015-03-02 00:15:00	612.917945
13420_1_0_1_1	2015-03-02 00:15:00	493.752842

Comprobamos que la fecha y la hora están en un formato con el que no podemos trabajar de forma independiente, así que podemos empezar a decidir algunas de las actuaciones que podremos hacer sobre las variables que tenemos, empezaremos por separar la fecha de la hora,

Otra de las cosas que podremos hacer es convertir la hora a un formato, con el cual podamos trabajar, tendríamos que trabajar sobre una única unidad de tiempo.

Tenemos datos desde la fecha '2015-02-28 23:00:00' hasta '2015-04-30 23:45:00', si tenemos en cuenta solo el día del mes, tenemos datos de 61 días, que corresponden a los meses de Marzo y Abril y una hora del mes de Febrero

5.4.3.3 Preparación de los datos

Con la observación de los datos concluimos que tenemos que transformarlos para así poder trabajar con ellos.

Convertir la hora a un formato, con el cual podamos trabajar, se decidió pasarlo todo a minutos, y así trabajar sobre la misma unidad de tiempo. Tenemos en cuenta los valores de las horas y los minutos, ya que en la observación de los datos vimos los segundos

siempre son 00. Así que añadimos una variable más que sea los minutos a los que corresponde cada una de las *horas:minutos* que tenemos.

*Minutos = int(minutos) + int(horas)*60*

Tenemos datos desde la fecha '2015-02-28 23:00:00' hasta '2015-04-30 23:45:00', como ya hemos separado la hora del formato, solo tenemos las fechas, para realizar el estudio vamos a tener en cuenta solamente los días de la semana, es decir, de Lunes a Domingo, con lo cual deberemos trasformar los datos para añadir una variable que sea que día de la semana corresponde cada una de las fechas del mes que tenemos.

Tenemos más de 5 millones de registros para trabajar, antes de empezar a modelar y explicar los datos, hemos realizado un proceso para separar ciertos datos de nuestro *dataset*, con el que vamos hacer el entrenamiento un 76% de la información y el resto 26% para hacer la validación del modelo. Para ello hemos tenido en cuenta desde el día 28-02 hasta el 15-04 incluidos como datos de entrenamiento y los últimos 14 días del mes de Abril para evaluación, en estos 14 días tenemos 2 semanas de datos, es decir 2 Lunes, 2 Martes, etc.

Todo ello está reflejado en el proceso de Python “preparacionRoma.py”

5.4.3.4 Modelado

Una vez preparados los datos, vamos a clasificar la información que tenemos en 2 grupos, Aglomeración SI/NO, para ello vamos a tener en cuenta las variables Grid, Día de la semana, Minutos y Presencia.

Dentro de todas las posibles opciones que hemos estudiado, hemos decidido para explicar el problema, agrupar los datos de presencia del fichero de entrenamiento en día de la semana, grid y minutos. Así que tendremos por cada día de la semana todos los grid y a su vez agrupado por minutos.

Con esta agrupación vamos a calcular la media y la desviación típica de la presencia. Así que vamos a obtener por día de la semana, grid y minuto la media aritmética de la presencia y la desviación típica de esa agrupación.

Una vez con esta información vamos a calcular para cada registro de nuestro fichero de entrenamiento y con la media y la desviación estándar correspondiente del día de la semana, grid y minuto calculado anteriormente que diferencia existe con los datos de presencia, es decir que Factor hace que se cumpla esta formula

*Presencia = Media + Factor * Desviación.*

Para ello simplemente despejaremos de la ecuación la variable Factor y lo calcularemos para cada uno de los registros de nuestro fichero de entrenamiento.

A partir de este punto teníamos dos posibilidades para el cálculo de las aglomeraciones, realizar ese cálculo teniendo en cuenta el día de la semana, el grid y el minuto en el que ocurre, o tomar el día de la semana y el grid. Nos pareció más útil usar la segunda de las posibilidades para así poder ver para cada día de la semana y grid, cual eran los momentos del día con aglomeración, en ese gris y ese día de la semana. Para lo cual, agrupamos nuestros factores, calculados anteriormente, por día de la semana y grid y nos quedamos con el de mayor valor.

Ahora tenemos el máximo factor para cada día de la semana y grid que hace que cuando la formula anterior se cumpla o la presencia sea superior, podemos concluir que hay una presencia superior al resto de registros del día y grid.

El resultado de este proceso lo que nos dio es una clasificación de Aglomeración SI/NO para cada uno de los registros de nuestro fichero de entrenamiento, teniendo en cuenta que los estamos haciendo por día de la semana y grid.

Si no tuviéramos en cuanta nada más que los datos, sin agrupar por dia de la semana y grid, el factor máximo que nos explicaría si tenemos o no Aglomeraciones es **2.267672.Es**

decir que cualquier valor de la variable presencia que iguale o supere la formula Media+ 2.267672 * Desviación, entendemos que nos explicaría que hay aglomeración.

Como sí que hemos usado la opción de tener en cuenta el día de la semana y grid, ya que entendemos que no todos los días ocurren las mimas cosas y no es lo mismo un grid céntrico o con más puntos de interés que otro en las afueras, un ejemplo del resultado para un día como el sábado serían:

GRID	DIA	FECHA	HORA	PRESENCIA	FACTOR	AGL
3423_3_3_2_1	Sábado	2015-02-28	23:00:00	2240.9353593	1.7046164565000002	NO
3418_2_2	Sábado	2015-02-28	23:00:00	1948.39062293	1.6871450585900001	NO
3420_2_0_3_2	Sábado	2015-02-28	23:00:00	1623.5030245	1.3324712104600001	SI
3419_1_1_3	Sábado	2015-02-28	23:00:00	1566.12298308	1.8228420749700001	SI
3420_3_3_1_1	Sábado	2015-02-28	23:00:00	3810.43988933	0.9782713596490000	NO
3419_3_1_2	Sábado	2015-02-28	23:00:00	909.654161953	1.9093891867700001	NO

Como vemos en el ejemplo no por tener un valor más alto la variable presencia querrá decir que hay aglomeración, ya que va a depender de que grid es el que estamos estudiando.

Todo ello está reflejado en el proceso de Python “modeladoRoma.py”

5.4.3.5 Evaluación

Una vez obtenidos los resultados de la formula seleccionada para la predicción, hicimos un breve estudio de la cantidad de positivos y negativos encontrados y nos dimos cuenta que el modelo nos ofrecía una cantidad superior de positivos de lo que nosotros queríamos obtener.

Realizando diferentes comprobaciones de registros al alzar, comparando diferentes días de la semana, nos dimos cuenta que al estar usando un numero decimal en la variable presencia, la cantidad de decimales que tomáramos afectaba al resultado de la clasificación, por lo que después de varios cálculos usando distinto número de decimales, optamos por quedarnos con 6 decimales, para la variable presencia.

Una vez ajustado y clasificados los datos de nuestro fichero de entrenamiento, realizamos los cálculos usando el 26% de los datos que reservamos al principio de trabajo para realizar las comprobaciones, con datos que no se habían usado en el cálculo del factor.

Todo ello está reflejado en el proceso de Python "evaluacionRoma.py"

Para comprobar que el modelo cumple con lo deseado, buscamos la situación del estadio de futbol donde juega el equipo de la Roma y la Lazio, el Estadio Olímpico, y lo situamos en el grid de los datos usados. Buscamos la hora y los días, dentro del rango de días que de los que tenemos datos, en los que se disputo algún partido en ese estadio.

COMPETICION	FECHA/HORA	EQUIPO	EQUIPO
Liga	**02/03/2015 20:45**	Roma	Juventus
	16/03/2015 21:00	Roma	Sampdoria
	04/04/2015 12:30	Roma	Napoli
	19/04/2015 15:00	Roma	Atalanta
	09/03/2015 19:00	Lazio	Fiorentina
	22/03/2015 20:45	Lazio	Verona
	12/04/2015 15:00	Lazio	Empoli
	26/04/2015 15:00	Lazio	Chievo
	29/04/2015 20:45	Lazio	Parma
COPA	**04/03/2015 20:45**	Lazio	Napoli
EUROPA LEAGUE	**19/03/2015 19:00**	Roma	ACF Fiorentina

En base a esta información, y sabiendo que el grid de la Roma son 3420_0_0_3_2_3 y 3420_0_0_3_2_2 y que los circundantes son 3420_0_3_0_1_0, 3420_0_3_0_1_1 ,3420_0_0_3_2_1 ,3420_0_0_3_2_0, realizamos los cruces con nuestros datos para comprobar si en un evento de las características de un partido de futbol nuestro modelo era capaz de darnos información, sobre si existía concentración de gente fuera de lo habitual y los resultados fueron:

Liga	**02/03/2015 20:45**	Roma	Juventus

GRID	HORA	PRESENCIA
3420_0_0_3_2_2 (estadio)	22:30:00	2460.89724097
3420_0_3_0_1_0 (cercanías)	22:30:00	1943.97076086
3420_0_3_0_1_1 (cercanías)	22:00:00	2378.56887464
3420_0_0_3_2_1 (cercanías)	21:45:00	578.277375464
3420_0_0_3_2_0 (cercanías)	21:45:00	654.631353811

Liga	**12/04/2015 15:00**	Lazio	Empoli

GRID	HORA	PRESENCIA
3420_0_3_0_1_0 (cercanías)	15:15:00	2055.06992598
3420_0_3_0_1_1 (cercanías)	15:15:00	1362.19886846
3420_0_0_3_2_1 (cercanías)	15:00:00	409.086122835
3420_0_0_3_2_2 (estadio)	14:30:00	3042.91249289
3420_0_0_3_2_0 (cercanías)	14:30:00	472.206129229
3420_0_0_3_2_3 (estadio)	14:30:00	5376.22252065

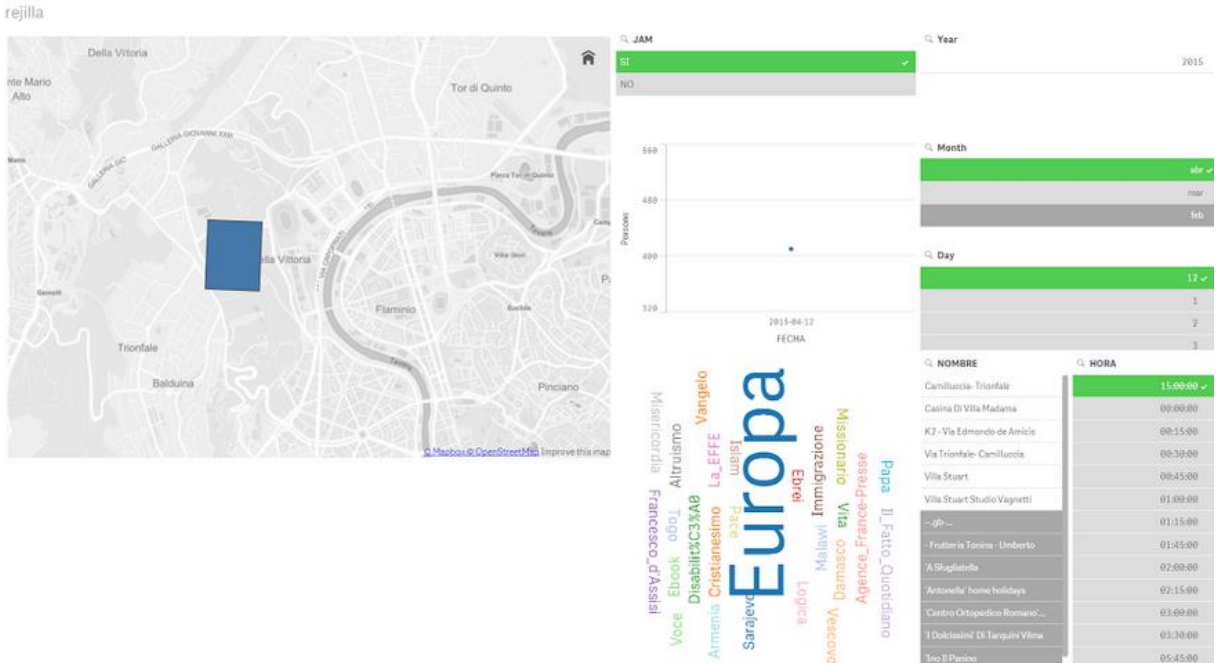
COPA	**04/03/2015 20:45**	Lazio	Napoli

GRID	HORA	PRESENCIA
3420_0_0_3_2_0 (cercanías)	21:00:00	418.088524755
3420_0_0_3_2_3 (estadio)	21:15:00	4576.85259899
3420_0_3_0_1_0 (cercanías)	20:00:00	1983.71587307
3420_0_0_3_2_1 (cercanías)	21:45:00	442.779999059
3420_0_0_3_2_2 (estadio)	21:45:00	2602.76379779
3420_0_3_0_1_1 (cercanías)	22:00:00	1169.31752219

EUROPA LEAGUE	**19/03/2015 19:00**	Roma	ACF Fiorentina

GRID	HORA	PRESENCIA
3420_0_3_0_1_1 (cercanías)	20:15:00	1333.465108
3420_0_0_3_2_1 (cercanías)	18:30:00	472.159156395
3420_0_0_3_2_0 (cercanías)	17:45:00	436.059599358
3420_0_3_0_1_0 (cercanías)	18:00:00	1941.74156873
3420_0_0_3_2_2 (estadio)	19:45:00	3006.47946076
3420_0_0_3_2_3 (estadio)	19:45:00	4192.48019015

Comprobamos que en todos los casos, al menos una de las antenas, de las dos en las que se dividen los datos del estadio Olímpico, aparecen como una concentración de gente, y en momentos anteriores o posteriores al partido, existen altas concentraciones de presencia en los alrededores, fruto de la entrada o salida de personas al estadio, esto nos da conocimiento del movimiento de las llamadas antes y después de un evento.



Aglomeración en Lazio-Empoli

Aglomeración en Roma-ACF Fiorentina

Como se puede observar, en los dos ejemplos visuales existe una concentración de gente en el estadio de La Roma. Sin embargo, se ve que en el segundo caso también existe una concentración en puntos alejados a las 19.00 horas. Las concentraciones se calculan independientemente para cada grid sin tener en cuenta el de al lado, es decir, si dos grids continuos tienen un tamaño muy diferente esto no se tiene en cuenta ya que lo que interesa es saber el patrón de cada grid por separado. Esto es porque un grid muy grande es habitual que registre mucha presencia pero si siempre mantiene un patrón en el que registra mucha presencia, debido al tamaño del grid en la mayoría de los casos, no quiere decir que haya una concentración. Será solo en aquellos casos en los que registre picos inusuales donde habrá concentración. Cada cálculo de si existe concentración o no se determina únicamente con los datos de su grid para identificar patrones anormales en ese grid. Si se tuviese en cuenta el tamaño del grid y la presencia, es decir, si se trabajase con la densidad, se estaría calculando los puntos de aglomeración de todos los grids conjuntamente pero no los patrones de cada uno de ellos por separado. Por poner un ejemplo gráfico, a un comercio situado en un grid determinado le interesa saber la tendencia en la que los picos de gente que se concentra son inusuales, y no el cálculo de

todos los grids conjuntos que indicaría los puntos de la ciudad donde se sufren más aglomeraciones. Un grid en el que se acumulasen 100 personas todos los días a las 15.00 horas y otro en el que se acumulasen 10000 personas habitualmente a las 15.00 horas, teniendo ambos el mismo tamaño, si se tuviese en cuenta la densidad provocaría que en el primer grid nunca habría aglomeración, cuando en realidad lo único que significa es que en el segundo grid habitualmente se concentra más gente. A un negocio o empresa que esté en el grid donde habitualmente se concentran 100 personas, le interesa saber cuándo se romperá esa tendencia, independientemente de lo que suceda en los grids colindantes, dato que por supuesto también influirá en cómo se concentra la gente en su grid.

5.4.3.6 Despliegue

Un de las salidas de proyecto es un cuadro de mando con la información del movimiento de la gente, así como de los tweets que se están generando en esas zonas, para la mejor gestión de los servicios y orden públicos, otra posible salida sería una App para dispositivos móviles con la información de puntos calientes de la ciudad, información de medios de transporte para salida de eventos e información comercial.

Nosotros para el alcance de este proyecto hemos realizado la primera de las opciones, un cuadro de mando aunando por un lado la información con la presencia de concentraciones altas de llamadas en base al modelo en un grid, fecha y hora seleccionada y añadiendo los hashtags de los tweets que se están originando en ese grid y en ese instante de tiempo, los puntos de interés que hay en ese grid, y los datos de los comercios de tenemos en ese grid.

Entre otras posibilidades nos permite:

1. Dar salida a grandes volúmenes de personas por diferentes vías de transporte, tanto público como privado gestionando los mismos en base a previsiones o análisis en tiempo real de cómo se concentran las personas
2. Prever seguridad y auxilio ante la previsión de concentraciones de personas
3. Servicios de limpieza o, auxilio (bomberos, ambulancias…)

4. Promover y activar la economía, ya que se ofrece información a los usuarios de eventos que están sucediendo en su ciudad, los sitios de los que se está hablando y por qué, actividades de ocio etc.
5. Ayuda a las empresas y comercios de los lugares de concentración, o bien para la gestión de sus ofertas y stocks, como la previsión de incidentes.

5.4.4 Desarrollo de cuadro de mando en Qlik Sense

Al principio se pensó en la utilización de otro producto de Qliktech para la realización del cuadro de mando, Qlikview Desktop Edition, con el que incluso se llegó a desarrollar, pero se descartó por la dificultad a la hora de pintar el mapa proporcionado en los datasets utilizados. El producto que se eligió para la realización del cuadro de mandos ha sido QlikSense producto más enfocado al usuario final.

Para ganar agilidad en las recargas se ha usado Qlikview Desktop para la creación de los ficheros QVDs, fichero propietario de QlikTech que agiliza las cargas de los datos al mismo tiempo que los comprime.

Muestra donde hay aglomeración para la fecha indicada

Una vez construida la primera capa de obtención y modelización del modelo de datos se crean los ficheros QVDs justo con el modelo de datos que queremos y pasamos a la carga de los datos y los mapas.

Modelo de datos

6 EVOLUCIONES FUTURAS

Hemos querido plantearnos cómo y en qué campos podríamos seguir mejorando nuestra idea y qué otras posibles fuentes de datos podríamos incorporar para hacer más inteligente el uso de los recursos y ayudar a la gestión de la ciudad.

Dentro de los posibles usos que el sistema ofrece estarían:

1. Control de recursos energéticos (encendido y apagado de luces en función de la concurrencia de ciudadanos.
2. Apertura de semáforos en función del tráfico.
3. Pasos de cebra inteligentes que detectan cuando un ciudadano está cruzando o quiere cruzar para abrir o cerrar semáforos.
4. Aumento de antenas de telefonía.
5. Control del ruido.

Para poder seguir mejorando la información y poder tener modelos predictivos, podríamos añadir nuevas fuentes de información, tanto pública como privada, entre otras, podrían estar:

1. Incidencias de tráfico.
2. Uso de trasporte público.
3. Sensores de detección de presencia.
4. Consumo eléctrico.
5. Datos de transacciones bancarias.
6. Facebook, Instagram.
7. Datos de telefonía en tiempo real.
8. Agenda de Eventos musicales, deportivos.

7 GESTIÓN DEL TIEMPO

7.1 Planificación y gestión temporal del proyecto

7.1.1 Definición del problema

Los integrantes del grupo de trabajo que desarrollamos el presente proyecto, tuvimos la oportunidad de trabajar juntos en una asignatura para investigar y definir un problema al que dar solución mediante los conocimientos adquiridos durante el periodo de duración del máster. De esta forma, se había hecho una labor de investigación previa sobre las posibilidades a la hora de intentar predecir cómo se concentra la gente dentro de las poblaciones cuando la concentración no estaba prevista, por qué y qué cantidad de gente se concentra, así como los posibles beneficiarios de obtener dicha información y cómo ayudaría la misma al desarrollo del concepto de *Smart Cities*.

Debido a ello, la labor de definición del problema nos llevó apenas una semana, si bien es verdad que la misma ha ido variando a medida que se desarrollaba el proyecto y se conocían nuevas fuentes de datos y las facilidades de acceso a la información.

Además, la idea inicial era desarrollar el proyecto acotando el ámbito de actuación de forma que pudiese desarrollarse un piloto en una ciudad que permitiese reflejar claramente el problema. En una primera aproximación se pensó en Madrid ya que se trata de la capital de España, tiene un censo de población muy elevado respecto al resto de poblaciones en España, posee una gran cantidad de negocios e instituciones que se verían beneficiados con el desarrollo del proyecto, es una ciudad que recibe una gran cantidad de turistas y posee una gran actividad en las redes sociales. Sin embargo, poco después conocíamos el concurso de Telecom Italia '*Big Data Challenge 2015*' que ponía a disposición de los participantes una gran cantidad de datos que encajaban a la perfección con la misión de nuestro proyecto. El problema seguía siendo el mismo, la predicción de aglomeraciones no controladas, pero los datos que el concurso ofrecía para que los participantes hicieran uso de ellos de una forma innovadora y diferencial, eran mucho más adecuados y enriquecedores de los que se disponía para Madrid, ya que por ejemplo los

datos de antenas telefónicas son casi imposibles de obtener y en el caso de Telecom Italia se ofrecían como Open Data de forma agregada para el concurso.

No solo los datos de las antenas telefónicas eran importantes y simplificaban en gran medida la obtención de una posible solución al problema, sino que además se tenía un acceso ilimitado a la API de Twitter para obtener información geolocalizada.

De esta forma, la definición del problema varió de una forma significativa en cuanto a la resolución del mismo. No obstante, la misión de predecir aglomeraciones no controladas seguía siendo el *core* del proyecto. De esta forma, la visión inicial del problema y su solución pasaron por un proceso de madurez de la idea y cambios hasta llegar a la definición final del problema:

- Definición inicial: tratar de predecir aglomeraciones y concentraciones no controladas en base a información geolocalizada obtenida a través de redes sociales, centrándose en un primer momento en la ciudad de Madrid.
- Definición final: tratar de predecir aglomeraciones y concentraciones no controladas en base a datos de antenas telefónicas, centrándose en un primer momento en la ciudad de Roma y apoyándose en información geo-localizada de redes sociales para tratar de explicar por qué se aglomeran las personas en un punto.

7.1.2 Recopilación de información

La recopilación de información, acceso a fuentes de datos e investigación y validación de la información recopilada compone uno de los hitos más importantes del proyecto, ya que el hito principal, el de la definición del problema, se ve en gran medida afectado por las posibilidades de obtención de datos para dar solución al mismo.

De esta forma, se invirtió algún tiempo durante las primeras semanas a recopilar información para evaluar las posibilidades de obtener una solución razonable al problema

definido. Como se ha comentado, en un primer momento se evaluaron los recursos e información disponibles para la ciudad de Madrid, en su mayoría información Open Data. Asimismo, se evaluó cómo acceder a la API de Twitter y ver qué cantidad de información podía obtener. Dicha API tiene unos límites a la hora de generar consultas y obtener información, tanto por ventanas temporales (15 minutos), como por peticiones para un recurso determinado dentro de esa ventana, dependiendo del recurso al que se intente acceder. Además, se comprobó que la gran mayoría de tweets no están geolocalizados, con lo que se despreciaría una gran parte de información para calcular cómo se aglomeran las personas.

A la vez que se recopilaba información para dar solución a la definición inicial del problema centrándonos en la ciudad de Madrid, ya se sabía del concurso '*Big Data Challenge 2015*' de Telecom Italia y se barajaba la posibilidad de redefinir el ámbito de actuación del piloto y la forma en que se iba a dar solución al problema si la información proporcionada por el concurso se adaptaba en mejor medida a la solución del problema.

Poco después de comenzar el hito, a las dos semanas, el concurso abrió la suscripción al mismo y la plataforma de datos a los concursantes. Tras el estudio de los *datasets* disponibles, así como de la cantidad y calidad de la información, se decidió centrar el ámbito del proyecto para la capital de Italia, Roma, y redefinir las líneas que se seguirían para dar solución al problema en base a la nueva información disponible.

Twitter pasaría a un segundo plano, dando apoyo a las predicciones de forma que se pudiese dotar de un significado semántico a las concentraciones y aglomeraciones. Los datos de las antenas telefónicas constituirían la fuente de datos principal para calcular cómo y dónde se concentran las personas y tratar de predecir cómo lo harían en el futuro.

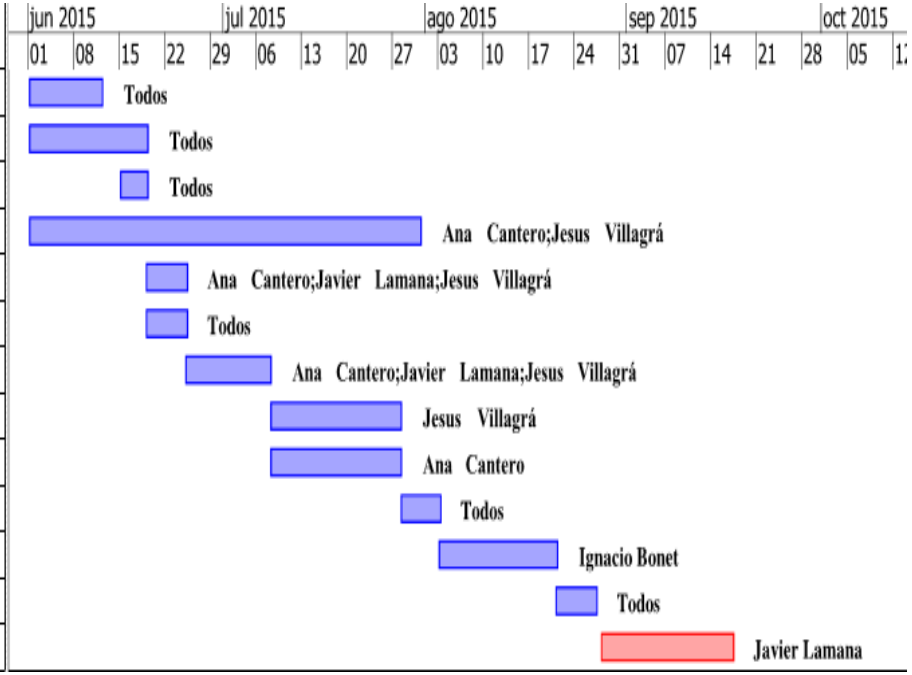
7.1.3 ***Desarrollo del piloto***

El desarrollo del piloto compone el eje principal del proyecto. Para ello se ha invertido la mayor parte del tiempo y de los recursos físicos.

En total, se han invertido unas nueve-diez semanas en elaborar el producto final contemplado en el alcance del Plan de Acción.

7.1.4 ***Resumen.***

	Nombre	Duración	Inicio	Terminado	Nombres del Recurso
1	Definicion del problema	10 days	1/06/15 ...	12/06/15...	Todos
2	Busqueda de Dataset	15 days	1/06/15 ...	19/06/15...	Todos
3	ValidacióndelProblema/Solución	5 days	15/06/15...	19/06/15...	Todos
4	Entrevistas de validación	45 days	1/06/15 ...	31/07/15...	Ana Cantero;Jesus Villagrá
5	Analisis de los datos	5 days	19/06/15...	25/06/15...	Ana Cantero;Javier Lamana;Jesus Villagrá
6	Modelos de negocio	5 days	19/06/15...	25/06/15...	Todos
7	Extracción de datos y ETL	10 days	25/06/15...	8/07/15 ...	Ana Cantero;Javier Lamana;Jesus Villagrá
8	Procesos de DataMining	15 days	8/07/15 ...	28/07/15...	Jesus Villagrá
9	Procesos de Tweets	15 days	8/07/15 ...	28/07/15...	Ana Cantero
10	Validación de los resultados	5 days	28/07/15...	3/08/15 ...	Todos
11	Representacion del Modelo	15 days	3/08/15 ...	21/08/15...	Ignacio Bonet
12	Optimizacion	5 days	21/08/15...	27/08/15...	Todos
13	Documentación	15 days	28/08/15...	17/09/15...	Javier Lamana



8 INDICADORES

Entre los posibles indicadores que nos indiquen que estamos siguiendo la estrategia marcada hemos considerado los siguientes:

- **Tasa Acierto Aglomeración** = Predicción / Total. Sería la tasa de acierto de nuestra herramienta en sus predicciones a la hora de detectar una aglomeración.
- **% Acierto Volumen** = Cantidad de gente pronosticada / Total Personas. Nos mide la bondad de la predicción en cuanto a la cantidad de gente.
- **Ratio Disminución Incidentes**: Incidentes actuales / Incidentes año anterior. Es un indicador que nos servirá para validar otros ya que si disminuyen las manifestaciones no controladas deberían disminuir los incidentes.
- **Tasa Acierto Motivo Aglomeración**. Del total de aglomeraciones, en cuantas se ha acertado el motivo de que se haya producido.
- **Número de alianzas tecnológicas**. Número de nuevas alianzas tecnológicas (anual).
- **Número de ciudades** usando herramienta.
- **Usuarios móviles**. Aumento de ciudadanos/ usuarios de a pie que usan la herramienta.
- **Presencia geográfica**. Expansión en nuevas ciudades o países donde está presente la herramienta.
- **Variación Inversión Ayuntamientos**: mide el aumento/disminución de la inversión de los ayuntamientos en nuestra herramienta.

9 *PLAN ECONÓMICO-FINANCIERO.*

El siguiente documento refleja el plan económico-financiero para el primer año de vida del proyecto:

Año 1 (€)													
Cuenta de Resultados	Ene	Feb	Mar	Abt	May	Jun	Jul	Ag	Sep	Oct	Nov	Dic	Total
Ventas	5.559	6.160	9.315	9.916	10.517	11.419	12.621	12.170	12.621	12.921	15.025	18.571	136.820
Compras	150	60	60	60	60	60	120	60	60	60	60	90	901
Alquileres	601	601	601	601	601	601	601	601	601	601	601	601	7.212
Gastos Personal	8.083	8.083	8.083	8.083	8.083	8.083	13.312	8.083	8.083	8.083	8.083	13.312	107.460
Otros Gastos	300	300	300	300	300	300	300	300	300	300	300	300	3.606
Intereses Banco	118	118	118	118	118	118	118	118	118	118	118	1.308	118
Amortizaciones	601	601	601	601	601	601	601	601	601	601	601	601	7.212
Total	-4.177	-3.604	-449	151	752	1.654	-2.432	2.405	2.856	3.156	5.260	3.547	9.119
Presupuesto de Tesoreria													
Tesoreria	0	23.319	20.321	20.479	19.885	21.244	23.506	20.327	23.340	26.803	29.214	35.081	0
Cobros	5.559	6.160	9.315	9.916	10.517	11.419	12.621	12.170	12.621	12.921	15.025	18.571	136.820
Capital	30.050												30.050
Prestamo	30.050												30.050
Total	65.660	29.479	29.637	30.395	30.403	32.664	36.127	32.498	35.961	39.725	44.240	53.653	196.921
Pagos													
Inversión y mobiliario	36.060												36.060
Compras	150	60	60	60	60	60	120	60	60	60	60	90	900
Gastos de Personal	5.228	7.632	7.632	8.985	7.632	7.632	14.213	7.632	7.632	8.985	7.632	12.861	103.704
Alquiler Local	601	601	601	601	601	601	601	601	601	601	601	601	7.212

Otros Gastos	300	300	300	300	300	300	300	300	300	300	300	300	3.606
Devol. Prestamo	0	563	563	563	563	563	563	563	563	563	563	563	6.200
Total	42.341	9.158	9.158	10.510	9.158	9.158	15.799	9.158	9.158	10.510	9.158	14.416	157.684
Saldo Tesoreria	23.319	20.321	20.479	19.885	21.244	23.506	20.327	23.340	26.803	29.214	35.081	39.236	39.236
Balance													
Activo													
Inmovilizado	36.060	36.060	36.060	36.060	36.060	36.060	36.060	36.060	36.060	36.060	36.060	36.060	36.060
Amortización	-601	-1.202	-1.803	-2.404	-3.005	-3.606	-4.207	-4.808	-5.409	-6.010	-6.611	-7.212	-601
Caja/Bancos	23.319	20.321	20.479	19.885	21.244	23.506	20.327	23.340	26.803	29.214	35.081	39.236	23.319
Total Act.	58.778	55.180	54.736	53.542	54.300	55.960	52.181	54.592	57.455	59.265	64.531	68.084	58.778
Pasivo													
Capital	30.050	30.050	30.050	30.050	30.050	30.050	30.050	30.050	30.050	30.050	30.050	30.050	30.050
Deudas/Banco	30.050	29.605	29.161	28.716	28.271	27.827	27.382	26.937	26.492	26.048	25.603	25.158	30.050
Proveedores	500	500	500	500	500	500	500	500	500	500	500	500	500
Resltado Año	-4.177	-7.781	-8.231	-8.079	-7.327	-5.673	-8.106	-5.700	-2.844	311	5.571	9.119	-4.177
Personal/SS/Hacienda	2.854	3.305	3.756	2.854	3.305	3.756	2.854	3.305	3.756	2.854	3.305	3.756	2.854
Total Pas.	**59.277**	**55.679**	**55.236**	**54.041**	**54.799**	**56.460**	**52.680**	**55.092**	**57.954**	**59.763**	**65.029**	**68.583**	**59.277**

10 ANEXOS

10.1 ***Entrevistas.***

10.1.1 ***Anexo II: Entrevista con el alcalde de Palencia.***

ENCUESTA PARA VALIDACIÓN DE HIPOTESIS

Código

- [x] Presentacion
- [ ] Contraste
- [ ] Piloto
- [ ] Venta

Fecha: 30/07/2015 Hora INICIO: 10:00 Hora FIN: 11:00

Lugar: Ayuntamiento de Palencia

Contacto: Alfonso Polanco

Posición: Alcalde

Empresa:
Dirección:
País:

GESTORES	PROFESIONALES	PROVEEDORES
Financiador S. S.	*Médico Publico*	*Prov. De Datos*
✓ *Gestor S. Pub.*	*Medico Privado*	*Prov. Software*
Gestor S. Priv.	*Conductores.*	*Prov. Tecnologia*
.	*Servicio de Limpieza.*	*Prov. Comunicación*
	Bomberos	

JUSTIFICACIÓN:

OBJETIVO:

VALIDACION DE HIPÓTESIS

SISTEMA DE ACTUAL DE CONTROL

1 *¿Señale lo tres principales problemas que a su juicio se enfrenta el sistema de control actualmente?*

1.1 Falta de anticipación
1.2 No contamos con los medios adecuados, en el momento necesario
1.3 Descordinación entre entidades
1.4
1.5

2 *¿Y en el futuro?*

2.1 Falta de recursos
2.2
2.3
2.4
2.5

3 *Valore de 1 a 10 la CALIDAD DE LA ASISTENCIA que recibe el ciudadano en una aglomeración 1. Muy deficiente 10. Sobr*

7

EFICACIA DEL SISTEMA

4 *¿Cuáles son, desde su punto de vista, las acciones para mejorar ACTUALMENTE la predición?*

4.1 Poder contar con los datos de movilidad de la gente
4.2 Tratar de anticiparse a las conversaciones de las redes sociales
4.3 Analizar lo que ocurrio en das señalados en años anteriores
4.4
4.5

5 *¿Cuáles son, desde su punto de vista, las TENDENCIAS FUTURAS para la mejorar el control de los servicios?*

EI

5.1 Recopilar de nuestro community manager la info de lo que se habla de Palencia
5.2 Poner sensores de presencia en las calles y plazas principales
5.3 Tratar de tener la informacin de los dispositivos moviles coneztados a Internet
5.4
5.5

EFICACIA DEL SISTEMA

6 *Valore de 1 a 10 el GRADO DE CONSUMO DE RECURSOS. 10 Bajo 5 Normal 1 desmesurado*

10

7 *¿Cuáles son, desde su punto de vista, las acciones para reducir/contener ACTUALMENTE los costes*

7.1 Ahora todo son recortes, pero no quitamos servicios de seguridad, ni de limpieza.
7.2 Acotamos las zonas de ocio a sitios concretos
7.3
7.4
7.5

8 *¿y las tendencias futuras relacionadas con el coste?*

8.1 Seria muy interesante poder adivinar las costumbres de la gente, para evitar desplazamientos y poder gestionar mejor las fuerzas de
8.2 seguridad y de salud no solo delayuntamiento, sino de la comunidad.
8.3
8.4
8.5

9 *5 primeras acciones que primero le recuerdan el control de las aglomeraciones*

9.1 Seguridad
9.2 Higiene
9.3 Respeto
9.4 Venta
9.5 Destrozos

10 *Indique el grado de importancia que para usted tiene las siguientes caracteriticas: 1.- Irrelevante 5,. Debe Estar 10.- Critico*

10.1	*Facil de usar*	8
10.2	*Diseño atractivo*	6
10.3	*Conectado al Sistema*	9
10.4	*Conectado al CRM*	9
10.5	*Portátil*	9
10.6	*Ligero*	
10.7		
10.8		
10.9		
10.10		

COMENTARIOS

11 *Observaciones generales*

Una de las cosas que mas valor teniene para los ciudadanos en una ciudad como Palencia, es que no se produzcan destrozos y se altere l
que esta ciudad tiene, con lo cual todo lo que ayude a ello seria perfecto.
Aquí principalmente los tumultos son de viandantes, el trafico es escaso y fluido, aunque en casos como los que hablamos,tambien es afec

10.1.2 *Anexo III: Entrevista con la gerente de Business Intelligence de Ferrovial*

ENCUESTA PARA VALIDACIÓN DE HIPOTESIS

Código

- [x] Presentacion
- [ ] Contraste
- [ ] Piloto
- [ ] Venta

Fecha: 21/07/2015 Hora INICIO: 10:00 Hora FIN: 11:00 **Lugar:** Ferrovial Corporación

Contacto: Ana Salgado
Posición: Gerente BI

Empresa: Ferrovial
Dirección: C/ Príncipe de Vergara
País: España

GESTORES	PROFESIONALES	PROVEEDORES
Financiador S. S.	*Médico Publico*	*Prov. De Datos*
Gestor S. Pub.	*Médico Privado*	*Prov. Software*
Gestor S. Priv.	*Conductores.*	*Prov. Tecnologia*
.	*Servicio de Limpieza.*	*Prov. Comunicación*
	Bomberos	✓ *Prov. Servicios*

JUSTIFICACIÓN:

OBJETIVO: Validación de idea para el PFM del Grupo de Trabajo 3, 'Smart Jam'

VALIDACION DE HIPÓTESIS

SISTEMA DE ACTUAL DE CONTROL

1 *¿Señale lo tres principales problemas que a su juicio se enfrenta el sistema de control actualmente?*

1.1 Existen pocas iniciativas para predecir aglomeraciones o indicar en tiempo real que se está acumulando gente
1.2 Gestión de los recursos públicos con base a aglomeraciones no eficiente
1.3 Información de difícil acceso: datos de telecomunicaciones
1.4
1.5

2 *¿Y en el futuro?*

2.1 Las redes sociales son un buen purto de partida, pero no existe una gran cantidad de información que pueda ser geolocalizada.
2.2
2.3
2.4
2.5

EFICACIA DEL SISTEMA

3 *Valore de 1 a 10 la CALIDAD DE LA ASISTENCIA que recibe el ciudadano en una aglomeración 1. Muy deficiente 10. Sobre.*

3

4 *¿Cuáles son, desde su punto de vista, las acciones para mejorar ACTUALMENTE la predición?*

4.1 Tratar la información de las compañías telefónicas, los datos de sensores, los datos del tráfico y las transacciones bancarias.

4.2 Para estas últimas, son importantes los acuerdos con empresas para poder acceder a la información.

4.3

4.4

4.5

5 *¿Cuáles son, desde su punto de vista, las TENDENCIAS FUTURAS para la mejora del control de los servicios?*

5.1 Cálculo de rutas de recogida de basuras con base a predicciones, previsión de servicios de salud y asistencia policial dependiendo

5.2 de la presencia que se espera en un punto, gestión y pedido de stock en comercios de una forma más certera, transporte público

5.3 optimizado y accesible en puntos de aglomeraciones.

5.4

5.5

EFICACIA DEL SISTEMA

6 *Valore de 1 a 10 el GRADO DE CONSUMO DE RECURSOS. 10 Bajo 5 Normal 1 desmesurado*

7 *¿Cuáles son, desde su punto de vista, las acciones para reducir/contener ACTUALMENTE los costes*

7.1 Los datos telefónicos, además de ser de difícil acceso, tienen un coste muy elevado. Es importante de alguna manera nacer de la mano

7.2 de las empresas telefónicas como una Start Up financiada por lanzaderas de las TELCOs.

7.3

7.4

7.5

8 *¿y las tendencias futuras relacionadas con el coste?*

8.1 Una vez que la idea tenga futuro y pueda prolongarse en el tiempo, la nube ofrece una gran cantidad de recursos que permiten a las

8.2 empresas desplegar en un corto espacio de tiempo sus productos reduciendo la inversión.

8.3

8.4

8.5

9 *5 primeras acciones que primero le recuerdan el control de las aglomeraciones*

9.1 Descongestión del tráfico, seguridad ciudadana, evitar incidentes y actos violentos, disponibilidad de transporte público y gestión de resic

9.2 de reisudos.

9.3

9.4

9.5

10 *Indique el grado de importancia que para usted tiene las siguientes caracteriticas: 1.- Irrelevante 5,. Debe Estar 10.- Critico*

10.1	*Facil de usar*	9
10.2	*Diseño atractivo*	8
10.3	*Conectado al Sistema*	10
10.4	*Conectado al CRM*	10
10.5	*Portátil*	9
10.6	*Ligero*	8
10.7		
10.8		
10.9		
10.10		

10.2 Anexo IV: Scripts

10.2.1 Obtención de tweets mediante consulta a API

```
##Recibe como entrada un fichero con las urls que llaman a la api y las palabras
definidas para buscar tweets con la misma

import json
import requests
from sys import argv
import os
import csv



def main():

    script, urls= argv
    txturls = open(urls)

    for row in csv.reader(txturls, delimiter='\t'):
        row[0]=str(row[0])
        row[0]=row[0].replace('[','').replace(']','').replace('"','')
        lee_Tweets(row[0], row[1])



def lee_Tweets(url,word):
    palabra=word
    contents = requests.get(url)
    jsonTweets=contents.json()
    prettyTweets=json.dumps(jsonTweets, sort_keys=True, indent=1, separators=(',', ':
'))
    save_local(prettyTweets, word)
    return prettyTweets

def save_local(Tweets, palabra):

    tweets=[]
    tweets=Tweets
    savePath='/home/ec2-user/tweets/Tweets-'+str(palabra)+'.csv'
    with open(savePath,'wb') as csvFile:
        csvFile.write(Tweets)

if __name__ == "__main__":
    main()
```

10.2.2 Pasos previos a la ETL en Python

10.2.2.1 Parseo puntos de interés

```
#########################Python parseado de json a linea por
documento################################

## Existencia de varios ficheros de puntos de interés: alojamiento, atracciones,
restaurantes etc ##

#######################################################################
##############################
json_str=open("C:/Users/ana/Desktop/JSON/SmartJam/bindingsAccommodation.json").r
ead()

lineas='json'
busca='}}'
lista=[]
for line in json_str:
    if line <> '\n':
        lineas=lineas+line
    else:
        line=''
        lineas=lineas+line

if busca in lineas:
    lineas=lineas.replace('}}', '}}\n')
    lineas=lineas.replace('json', '')
    num=len(lineas.split('\n'))-1
    for i in range(num):
        lista.append(lineas.split('\n')[i])

with open('C:/Users/ana/Desktop/bindingsAccommodation.json', 'wb') as f:
    for line in lista:
        f.write(line)
        f.write('\n')




import json
json_str=open("C:/Users/ana/Desktop/JSON/SmartJam/Attraction.json").read()

lineas='json'
busca='}}'
```

```
lista=[]
for line in json_str:
    if line <> '\n':
        lineas=lineas+line
    else:
        line=''
        lineas=lineas+line

if busca in lineas:
    lineas=lineas.replace('}}', '}}\n')
    lineas=lineas.replace('json', '')
    num=len(lineas.split('\n'))-1
    for i in range(num):
        lista.append(lineas.split('\n')[i])

with open('C:/Users/ana/Desktop/Attraction.json', 'wb') as f:
    for line in lista:
        f.write(line)
        f.write('\n')


json_str=open("C:/Users/ana/Desktop/JSON/SmartJam/bindingsPointOfInterest.json").re
ad()

lineas='json'
busca='}}'
lista=[]
for line in json_str:
    if line <> '\n':
        lineas=lineas+line
    else:
        line=''
        lineas=lineas+line

if busca in lineas:
    lineas=lineas.replace('}}', '}}\n')
    lineas=lineas.replace('json', '')
    num=len(lineas.split('\n'))-1
    for i in range(num):
        lista.append(lineas.split('\n')[i])

with open('C:/Users/ana/Desktop/bindingsPointOfInterest.json', 'wb') as f:
    for line in lista:
        f.write(line)
        f.write('\n')
```

```
json_str=open("C:/Users/ana/Desktop/JSON/SmartJam/bindingsProductOrService.json")
.read()

lineas='json'
busca='}}'
lista=[]
for line in json_str:
    if line <> '\n':
        lineas=lineas+line
    else:
        line=''
        lineas=lineas+line

if busca in lineas:
    lineas=lineas.replace('}}', '}}\n')
    lineas=lineas.replace('json', '')
    num=len(lineas.split('\n'))-1
    for i in range(num):
        lista.append(lineas.split('\n')[i])

with open('C:/Users/ana/Desktop/bindingsProductOrService.json', 'wb') as f:
    for line in lista:
        f.write(line)
        f.write('\n')


json_str=open("C:/Users/ana/Desktop/JSON/SmartJam/bindingsRestaurant.json").read()

lineas='json'
busca='}}'
lista=[]
for line in json_str:
    if line <> '\n':
        lineas=lineas+line
    else:
        line=''
        lineas=lineas+line

if busca in lineas:
    lineas=lineas.replace('}}', '}}\n')
    lineas=lineas.replace('json', '')
    num=len(lineas.split('\n'))-1
    for i in range(num):
```

```
        lista.append(lineas.split('\n')[i])

with open('C:/Users/ana/Desktop/bindingsRestaurant.json', 'wb') as f:
    for line in lista:
        f.write(line)
        f.write('\n')
```

10.2.2.2 Parseo Tweets

```
import json
import requests
from sys import argv
import os
import csv
from pprint import pprint
import numpy as np


ruta="C:/Users/Juancar/Desktop/Smart Jam/Scripts/Tweets_Python/Tweets-Radio.csv"

with open(ruta) as data_file:
    data = json.load(data_file)

exportar=[]


for tweet in range (len(data["items"])):
    recurso=''
    for item in range(len(data["items"][tweet]["entities"])):
        if item==0:
            recurso=(data["items"][tweet]["entities"][item].split('resource/')[1])
        else:
            recurso=recurso+';'+(data["items"][tweet]["entities"][item].split('resource/')[1])

exportar.append((data["items"][tweet]["user"],data["items"][tweet]["created"],data["it
ems"][tweet]["modified"],data["items"][tweet]["geometry"]["coordinates"][1],data["ite
ms"][tweet]["geometry"]["coordinates"][0],recurso,data["items"][tweet]["language"]))



rutaSave=ruta="C:/Users/Juancar/Desktop/Smart Jam/Scripts/Tweets_Python/Tweets-
Radio-Parsed.csv"

with open(rutaSave,'wb') as fileSave:
    writer = csv.writer(fileSave, delimiter='\t')
    writer.writerows(exportar)
```

10.2.3 ETL de las fuentes no parseadas

```
----------------------------------------ETL SMART JAM-----------------------------------------------
---------
-----------------------------------------------------------------------------------------------------
--

--Creación de la tabla inicial que lee los datos de presencua en cada grid de la malla-
Esta tabla de lectura deberá
--se external SIEMPRE ya que si borramos la tabla queremos que siga existiendo el
directorio con sus ficheros
--originales ya que estos no variarán puesto que son los datos de entrada.
--El delimitador del fichero se sabe que es  ',' (previamente me creo una tabla de un
solo campo tipo string para ver
--cómo son los datos y saber cuál es el delimitador y en cuanto sé el formato que tiene
elimino dicha tabla,
--SIEMPRE como external al ser solo de lectura).
drop table if exists aux_presence;

create external table aux_presence
(id_grid string, double_presencia double, date_unix_fecha string)
row format
delimited fields terminated by ','
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Entrada/TIM/Presence';


drop table if exists maestro_grid;

create external table maestro_grid
(id_grid string, double_superficie double)
row format
delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Entrada/TIM/Maestro_Grid';


--Transformaciones: conversión de la fecha en milisegundos de unix a un formato de
fecha adecuado, división del
--campo presencia entre 10. Para ello:
```

```
--1. Creamos la tabla que almacenará la información ya limpia y transformada. Si no
existe el directorio que le
--indiquemos, lo creará. La tabla no debe ser creada como external, ya que nos
interesa que si borramos la tabla
--se borren los datos que contiene el directorio, por si nos hemos equivocado haciendo
la ETL y queremos ahorrarnos
--tener que ir al directorio y borrar a mano los registros insertados.
drop table if exists presence;
--Creamos la tabla indicándole los campos que va a contener. Esta tabla puede
contener más campos que la auxiliar,
--los mismos, campos calculados a partir de la auxiliar... Eld elimitador en este caso
será \t.
create table presence
(id_grid string, double_presencia double, date_fecha timestamp, double_superficie
double, double_densidad double)
row format
delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Procesamiento/TIM/Presence';

--2. Insertamos los registros aplicándoles la transformación necesaria. Como el tiempo
que nos dan es en formato
--epoch unix y está en milisegundos habrá que dividirlo entre 1000 para pasarlo a
segundos, ya que la función
--en hive entenderá que le pasamos segundos.
insert into table presence
select
aux_presence.id_grid,
(double_presencia/10) as double_presencia,
cast(from_unixtime(bigint (date_unix_fecha/1000))as timestamp) as date_fecha,
double_superficie,
(double_presencia/double_superficie) as double_densidad
from aux_presence
left join maestro_grid
on maestro_grid.id_grid=aux_presence.id_grid;



--Para el dataset de negocios

drop table if exists aux_negocios;


create external table aux_negocios
(subject_id string, sign string , long double, lat double, kind string, area string)
```

```
row format delimited fields terminated by '\;'
lines terminated by '\n'
STORED AS TEXTFILE LOCATION 's3://bigcatachallenge/Entrada/Negocios'
tblproperties ("skip.header.line.count"="1");

drop table if exists negocios;
create table negocios
(str_id_negocio string, str_nombre_negocio string , float_long double, float_lat double,
str_tipo string)
row format delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE LOCATION 's3://bigcatachallenge/Procesamiento/Negocios';

insert into table negocios
select subject_id,
sign,
long ,
lat,
kind
from aux_negocios
where area<>''
order by subject_id;
```

10.2.4 ***ETL de las fuentes parseadas***

```
---------------------------------  PROCESAMIENTO DE FICHEROS JSON-------------------------------
----

--        Carga de json como tabla, limpieza y transformación de campos y
almacenamiento en S3           --



----------Carga del fichero de eventos JSON como tabla una vez lo tenemos en formato
una linea por doc
--Ademas, nos quedamos unicamente con los campos que nos interesa, le amos formato
y lo cargamos en una
--tabla de eventos final
add jar s3://elasticmapreduce/samples/hive-ads/libs/jsonserde.jar;

drop table if exists aux_eventos;
create external table aux_eventos
(
lugar string,
fdesde string,
```

```
fhasta string,
coorx string,
coory string,
direcci string,
id string
)
row format serde 'com.amazon.elasticmapreduce.JsonSerde'
WITH SERDEPROPERTIES ( 'paths'= 'Lugar,FDesde, FHasta, CoorX, CoorY, Direcci, id' )
LOCATION 's3://bigcatachallenge/Entrada/JSON_Eventos';

drop table if exists eventos;
create table eventos
(
id string,
lugar string,
direccion string,
fecha_comienzo timestamp,
fecha_fin timestamp,
coordx double,
coordy double
)
row format
delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Procesamiento/Eventos';

insert into table eventos
select
distinct(id),
lugar,
direcci as direccion,
cast(concat(split(fdesde,'-')[2],'-',(case when length(split(fdesde,'-')[1])=1 then
concat('0',split(fdesde,'-')[1]) else split(fdesde,'-')[1] end)
,'-',(case when length(split(fdesde,'-')[0])=1 then concat('0',split(fdesde,'-')[0]) else
split(fdesde,'-')[0] end),' ','00:00:00') as timestamp) as fecha_comienzo,
cast(concat(split(fhasta,'-')[2],'-',(case when length(split(fhasta,'-')[1])=1 then
concat('0',split(fhasta,'-')[1]) else split(fhasta,'-')[1] end)
,'-',(case when length(split(fhasta,'-')[0])=1 then concat('0',split(fhasta,'-')[0]) else
split(fhasta,'-')[0] end),' ','00:00:00') as timestamp) as fecha_fin,
cast(coorx as double) as coordx,
cast(coory as double) as coordy
from aux_eventos;
```

```
select
cast(concat(split(coordx,'.')[0],',',split(coordx,'.')[1]) as double) as coordx,
cast(concat(split(coordy,'.')[0],',',split(coordy,'.')[1]) as double) as coordy
from eventos;


----------------------------Creacion de la tabla auxiliar que carga JSON de puntos de
interes (solo campos que
 --interesan) y creacion de la tabla final de puntos de interes que contendra los campos
con el formato adecuado

drop table if exists aux_puntosInteres;
create external table aux_puntosInteres
(
id string,
tp_value string,
address_value string,
title string,
coordx string,
coordy string
)
row format serde 'com.amazon.elasticmapreduce.JsonSerde'
WITH SERDEPROPERTIES ( 'paths'= '_id, tp.value, address.value, title.value, lat.value,
long.value' )
LOCATION 's3://bigcatachallenge/Entrada/JSON_PuntosInteres';


drop table if exists PuntosInteres;
create  table PuntosInteres
(
id string,
tipo string,
direccion string,
nombre string,
coordx double,
coordy double
)
row format
delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Procesamiento/PuntosInteres';

insert into table PuntosInteres
select
split(id, "")[1] as id,
```

```
split(tp_value,'#')[1] as tipo,
address_value as direccion,
title as nombre,
cast(coordx as double) as coordx,
cast(coordy as double) as coordy
from aux_puntosinteres;
```

10.2.5 ETL creación de QVDs Qlikview.

```
SET ThousandSep=',';
SET DecimalSep='.';
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='#.##0,00 €;-#.##0,00 €';
SET TimeFormat='h:mm:ss';
SET DateFormat='DD/MM/YYYY';
SET TimestampFormat='DD/MM/YYYY h:mm:ss[.fff]';
SET MonthNames='ene;feb;mar;abr;may;jun;jul;ago;sep;oct;nov;dic';
SET DayNames='lun;mar;mié;jue;vie;sáb;dom';

Presencia_Calculada:
LOAD TExT(GRID) as GRIDRoma1.Name,
    //DIA,
    FECHA,
    Day(FECHA) as DIA,
    Month(FECHA) as MES,
    Year(FECHA) as ANIO,
    HORA as HORA,
    NUM(PERSONAS,'#,##0.00') as PERSONAS,
    FECHA2 as FECHA2,
    MINUTOS as MINUTOS,
    MEDIA as MEDIA,
    DESVIACION as DESVIACION,
    FACTOR as FACTOR,
    FACTORMAX as FACTORMAX,
    AGLOMERACION as AGLOMERACION
FROM
Ficheros\presencia_Calculada_?_FINAL.csv
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

//STORE Presencia_Calculada into Presencia_Calculada.qvd (qvd);

//Drop table Presencia_Calculada;
Concatenate
presencia_New_Aglomaracion:
```

```
LOAD TExT(GRID) as GRIDRoma1.Name,
    //DIA,
    FECHA as FECHA,
    Day(FECHA) as DIA,
    Month(FECHA) as MES,
    Year(FECHA) as ANIO,
    HORA as HORA,
    NUM(PERSONAS,'#,##0.00') as PERSONAS,
    FECHA2 as FECHA2,
    MINUTOS as MINUTOS,
    MEDIA as MEDIA,
    DESVIACION as DESVIACION,
    FACTOR as FACTOR,
    FACTORMAX as FACTORMAX,
    AGLOMERACION as AGLOMERACION

FROM
Ficheros\presencia_New_Aglomaracion.csv
(txt, codepage is 1252, embedded labels, delimiter is ',', msq);

//STORE presencia_New_Aglomaracion into presencia_New_Aglomaracion.qvd (qvd);


STORE Presencia_Calculada into Presencia_Calculada.qvd (qvd);

Tweet:
LOAD Text(@1) as GRIDRoma1.Name,
    @2 as FECHA_T,
    @3 as Palabro,
    Sum(@4) as repeticiones
    //º@4 as repeticiones
FROM
[Ficheros\c6476998-3979-424e-a20f-2621c6ea9b18-000000]
(txt, utf8, no labels, delimiter is '\t', msq)
Where len(@3)>0
Group by @3,@1,@2
;
STORE Tweet into Tweet.qvd (qvd);

POI:
LOAD FID,
    Join_Count,
    ID,
    TIPO,
    DIRECCION,
    NOMBRE,
```

```
    LAT,
    LONG_,
    TExT(ID_GRID) as GRIDRoma1.Name
FROM
Ficheros\PuntosInteres_GRID.txt
(txt, utf8, embedded labels, delimiter is '\t', msq);

STORE POI into POI.qvd (qvd);
```

10.2.6 ***Word Count***

```
drop table if exists aux_tweets;
create external table aux_tweets
(
user string,
date_created string,
date_modified string,
lat double,
long double,
dbpedia string,
language string,
id_grid string
)
row format
delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Procesamiento/Tweets/Tweets_GRID'
tblproperties ("skip.header.line.count"="1");


--Modificamos los campos de la fecha para que sean un timestamp, y hacemos un
explode de los registros dependiendo del numero de entidades de dbpedia que exista
-- para cada uno
drop table if exists tweets;
create table tweets as
select
user,
cast(concat((substring(date_created,0,10)), ' ', (substring(date_created,12,8))) as
timestamp) as date_created,
cast(concat((substring(date_modified,0,10)), ' ', (substring(date_modified,12,8))) as
timestamp) as date_modified,
lat,
long,
split(dbpedia, '\;') as dbpedia,
```

```
dbpedia_single,
language,
id_grid
from aux_tweets
lateral view explode(split(dbpedia, '\;')) asTable as dbpedia_single;

--Agregamos los tweets por grid y hora y contamos las repeticiones de cada palabra

drop table if exists tweets_wordCount;
create external table tweets_wordCount
(
id_grid string,
fecha string,
recurso string,
repeticiones int
)
row format
delimited fields terminated by '\t'
lines terminated by '\n'
STORED AS TEXTFILE
LOCATION 's3://bigcatachallenge/Procesamiento/Tweets/Word_Count';


insert into table tweets_wordCount
select
id_grid,
substring(date_created,0,13) as fecha,
dbpedia_single as recursi,
count(dbpedia_single) as repeticiones
from tweets
group by
id_grid,
substring(date_created,0,13),
dbpedia_single
order by repeticiones desc;
```

*10.2.7 **Análisis predictivos***

10.2.7.1 preparacionRoma.py

```
#Lo primero que vamos hacer en convertir la fecha a dias de la semana (Lunes,
Martes,Miercoles...)
# y convertimos la hora a minutos para poder trabajar mejor, con un formato númerico

import pandas as pd
```

```
import numpy
import os
import datetime
import csv
import fileinput

#ruta de I/O de los ficheros
datos_path = "C:\Personal\MASTER BI AND BIG DATA\Proyecto Fin de Master\Ficheros"

#Fichero original de información con datos de la ciudad Roma. Zonas divididas en
Rejillas, con información de las conexiones telefónicas móviles
#cada 15 minutos desde el 28 de Febrero al 30 de Abril

llamadas_file = "presencia.txt"

#Dividimos el fichero original, Uno para entrenamiento y otro para comprobar los datos
de entrenamiento
#El de train: del 28-02 al 15-04, el resto de datos para el new
llamadas_New="presencia_new.csv"
llamadas_Train="presencia_train.csv"

f=open(os.path.join(datos_path, llamadas_Train), 'wb')
fo=open(os.path.join(datos_path, llamadas_New), 'wb')

train = csv.writer(f)
new = csv.writer(fo)

train.writerow(["GRID","DIA","FECHA","HORA","PERSONAS","FECHA2","MINUTOS"])
new.writerow(["GRID","DIA","FECHA","HORA","PERSONAS","FECHA2","MINUTOS"])


for fila in fileinput.input([os.path.join(datos_path, llamadas_file)]):

    grid, personas, fechaN,hora=fila.split()

    year,month,day=fechaN.split("-")

    dicdias =
{'MONDAY':'Lunes','TUESDAY':'Martes','WEDNESDAY':'Miercoles','THURSDAY':'Jueves', \
    'FRIDAY':'Viernes','SATURDAY':'Sabado','SUNDAY':'Domingo'}
    anho = int(year)
    mes =  int(month)
    dia= int(day)

    fecha = datetime.date(anho, mes, dia)
```

```
    diasemana=dicdias[fecha.strftime('%A').upper()]

    horas, minutos, segundos =hora.split(":")
    tiempo=int(minutos) + int(horas)*60

    if mes==4 and dia>15:
        new.writerow([grid,diasemana,fechaN,hora,personas,fecha,tiempo])
    else:
        train.writerow([grid,diasemana,fechaN,hora,personas,fecha,tiempo])

f.close()
fo.close()
```

10.2.7.2 modeladoRoma.py

```
#Para poder saber si tenemos aglomeraciones,vamos a agrupar los datos del fichero de
train en dias de la semana, grid y minutos
# de esa manera tendre la información de las personas que hay por cada dia de la
semana, grid y minuto
# y calculo la media y la desviacion estandar

#Para calcular el factor uso la formula  factor=(Personas-media)/desviacion
# de est forma tengo el factor por dia de la semana, grid y minuto

import pandas as pd
import os
import osv

datos_path = "C:\Personal\MASTER BI AND BIG DATA\Proyecto Fin de Master\Ficheros"

llamadas_Train="presencia_train.csv"

llamadas_Train = pd.read_csv(os.path.join(datos_path, llamadas_Train), sep=',')

presence_min_loc = llamadas_Train.groupby(['DIA','GRID','MINUTOS'])['PERSONAS']
MEDIA=presence_min_loc.mean()
DESVIACION=presence_min_loc.std()


lista=open(os.path.join(datos_path, llamadasW_file),"wb")
factor_calculado = csv.writer(lista)
factor_calculado.writerow(["GRID" , "DIA",
"FECHA","HORA","PERSONAS","FECHA2","MINUTOS","MEDIA","DESVIACION","FACTOR"])


factor=0
```

```
for i in range(len(llamadas_Train)):
    DIA_MD=MEDIA[llamadas_Train.DIA[i]]
    MIN_MD=DIA_MD[llamadas_Train.MINUTOS[i]]
    md=float(MIN_MD[llamadas_Train.GRID[i]])

    DIA_DS=DESVIACION[llamadas_Train.DIA[i]]
    MIN_DS=DIA_DS[llamadas_Train.MINUTOS[i]]
    ds=float(MIN_DS[llamadas_Train.GRID[i]])


    factor=(llamadas_Train.PERSONAS[i]-md)/ds

    factor_calculado.writerow([str(llamadas_Train.GRID[i]),str(llamadas_Train.DIA[i]),
str(llamadas_Train.FECHA[i]),str(llamadas_Train.HORA[i]),str(llamadas_Train.PERSONA
S[i]),str(llamadas_Train.FECHA2[i]),str(llamadas_Train.MINUTOS[i]),str(md),str(ds),str(f
actor) ])

lista.close()

#una vez termino el calculo de cada factot, busco el maximo factor de cada grid
# de esta forma tenemos controlado las posibles aglomeraciones de un grid, con
independencia de la hora
#de esta forma, comprobando si las personas que estan en el grid es superior o igual al
max numero de personas que hemos
#calculado que podria haber, y añadimos una etiqueta 'SI' o 'NO' hay aglomeracion

llamadas_Train = pd.read_csv(os.path.join(datos_path, llamadas_Train), sep=',')
presence_max_fac = llamadas_Train.groupby(['DIA','GRID'])['FACTOR']
MAXIMOFACTOR=presence_max_fac.max()


llamadasW_file="presencia_Calculada_FINAL.csv"
lista=open(os.path.join(datos_path, llamadasW_file),"wb")

calculadafinal = csv.writer(lista)


calculadafinal.writerow(["GRID" , "DIA",
"FECHA","HORA","PERSONAS","FECHA2","MINUTOS","MEDIA","DESVIACION","FACTOR","FA
CTORMAX","AGLOMERACION"])

for i in range(len(llamadas_Train)):

    DIA_FACTOR=MAXIMOFACTOR[llamadas_Train.DIA[i]]
    factor=DIA_FACTOR[llamadas_Train.GRID[i]]
```

```
aglomaracion=float(llamadas_Train.MEDIA[i])+factor*float(llamadas_Train.DESVIACION[i
])


aglomeracion=float(llamadas_Train.MEDIA[i])+factor*float(llamadas_Train.DESVIACION[i
])
        #me quedo solo con 6 decimales
    pre="%.6f" % (int(llamadas_Train.PERSONAS[i]*1000000)/float(1000000))
    agl="%.6f" % (int(aglomaracion*1000000)/float(1000000))

        if float(pre)>= float(agl):
      calculadafinal.writerow([str(llamadas_Train.GRID[i]),str(llamadas_Train.DIA[i]),
str(llamadas_Train.FECHA[i]),str(llamadas_Train.HORA[i]),str(llamadas_Train.PERSONA
S[i]),str(llamadas_Train.FECHA2[i]),str(llamadas_Train.MINUTOS[i]),str(llamadas_Train.
MEDIA[i]),str(llamadas_Train.DESVIACION[i]),str(llamadas_Train.FACTOR[i]),factor,'SI'])
    else:
      calculadafinal.writerow([str(llamadas_Train.GRID[i]),str(llamadas_Train.DIA[i]),
str(llamadas_Train.FECHA[i]),str(llamadas_Train.HORA[i]),str(llamadas_Train.PERSONA
S[i]),str(llamadas_Train.FECHA2[i]),str(llamadas_Train.MINUTOS[i]),str(llamadas_Train.
MEDIA[i]),str(llamadas_Train.DESVIACION[i]),str(llamadas_Train.FACTOR[i]),factor,'NO'
])

lista.close()
```

10.2.7.3 evaluacionRoma.py

```
# Tenemos dos ficheros de lectura, por un lado el fichero de presencias calculado el
factor por dia, grid y minuto
# por otro lado el nuevo fichero con la información de grid, minuto y dia de la semana
# y la salida, será otro fichero con la informacion de las posibles aglomeraciones en el
dia , grid y minuto
# para ello buscamos en el fichero de train el factor maximo para ese dia de la semana,
grid y minuto, que será
#el mismo para todos los minutos, ya que es el factor maximo. y hacemos la
comprobacion con las personas que estan en el nuevo
#if presenciaFN.PERSONAS[i]>=aglomeracion: ----- SI aglomeración


# si quisieramos un calculo global, sin tener en cuenta los grid. el factor maximo es
# AGLOMERACION  2.267672

import pandas as pd
import csv
import os
```

```
import numpy

datos_path = "C:\Personal\MASTER BI AND BIG DATA\Proyecto Fin de Master\Ficheros"

presenciaF_file = "presencia_Calculada_FINAL.csv"
presenciaF = pd.read_csv(os.path.join(datos_path, presenciaF_file), sep=',')

presenciaFN_file = "presencia_New.csv"
presenciaFN = pd.read_csv(os.path.join(datos_path, presenciaFN_file), sep=',')

llamadasW_file= "presencia_New_Aglomaracion.csv"
lista=open(os.path.join(datos_path, llamadasW_file),"wb")

calculadanueva = csv.writer(lista)

calculadanueva.writerow(["GRID" , "DIA",
"FECHA","HORA","PERSONAS","FECHA2","MINUTOS","MEDIA","DESVIACION","FACTOR","FA
CTORMAX","AGLOMERACION"])


for i in range(len(presenciaFN)):
        dia=presenciaF[presenciaF.DIA==presenciaFN.DIA[i]]
        grid=dia[dia.GRID==presenciaFN.GRID[i]]
   temp=grid[grid.MINUTOS==presenciaFN.MINUTOS[i]]

        aglomeracion= numpy.array(temp)[0][7]+( numpy.array(temp)[0][10]*
numpy.array(temp)[0][8])

   pre="%.6f" % (int(presenciaFN.PERSONAS[i]*1000000)/float(1000000))
   agl="%.6f" % (int(aglomeracion*1000000)/float(1000000))

   if float(pre)>= float(agl):
      calculadanueva.writerow([str(presenciaFN.GRID[i]),str(presenciaFN.DIA[i]),
str(presenciaFN.FECHA[i]),str(presenciaFN.HORA[i]),str(presenciaFN.PERSONAS[i]),str(
presenciaFN.FECHA2[i]),str(presenciaFN.MINUTOS[i]),str(numpy.array(temp)[0][7]),str
(numpy.array(temp)[0][8]),str(numpy.array(temp)[0][9]),str(numpy.array(temp)[0][10
]),'SI' ])
   else:
      calculadanueva.writerow([str(presenciaFN.GRID[i]),str(presenciaFN.DIA[i]),
str(presenciaFN.FECHA[i]),str(presenciaFN.HORA[i]),str(presenciaFN.PERSONAS[i]),str(
presenciaFN.FECHA2[i]),str(presenciaFN.MINUTOS[i]),str(numpy.array(temp)[0][7]),str
(numpy.array(temp)[0][8]),str(numpy.array(temp)[0][9]),str(numpy.array(temp)[0][10
]),'NO' ])
```

10.2.8 ***Scraping***

```
import urllib
import re
import json

#para que la salida sea JSON
class UserEncoder(json.JSONEncoder):
    def default(self, obj):
        return obj.__dict__

class Salida(object):

    def __init__(self, ide, CX,CY,FDesde,FHasta,Lugar,Direcci):
        self.id = ide
        self.CoorX = CX
        self.CoorY = CY
        self.FDesde = FDesde
        self.FHasta = FHasta
        self.Lugar = Lugar
        self.Direcci = Direcci

# Abrimos un archivo para meter la info
archivo = open("c:\\personal\\actividadesculturales.txt", "w+")

#Hacemos los patrones
# regex es el que localiza las coordenadas
# regex2 localiza las direcciones

regex = 'ArchiMap.initMarker(.+?)<br'
pattern = re.compile(regex)
regex2 = '<li><h2><a href=(.+?)</p></li>'
pattern2 = re.compile(regex2)

dia = 1
mes = 1
serializeTotal=""

#recorremos los meses que necesitamos
while mes<=4:
    if mes in (1,3):
        diames=31
    elif mes==2:
        diames=28
```

```
    else:
        diames=30

    while dia<=diames:
        fecha =str(dia)+"%2F0"+str(mes)+"%2F2015"
        #leemos la url
        htmlfile=urllib.urlopen("http://www.turismoroma.it/eventi?data=" + fecha)
        htmltext=htmlfile.read()

                #para cada valor de las coordenadas,
        for m in re.finditer(pattern,htmltext):
            #print m.group(1).split('<a')[0].split(',')[1]
            #print m.group(1).split('<a')[0].split(',')[2]

            #print m.groups()
            id = str(m.group(1).split('<a href=')[1].split('>')[0]).replace ("\'"," ")
            id = id.replace ("\\"," ").split("=")

            #por cada valor de las direcciones
            for s in re.finditer(pattern2,htmltext):
                id2=str(s.group(1).split('>')[0]).replace("\'"," ").split("=")

#comparo con la direccion y escribo si existen coordenadas para la direccion

                if int(id[2][2:])==int(id2[3][2:]):
                    listas = s.group(1).split('>')
                    # id, y las 2 coordenadas
                    ident=str(m.group(1).split('<a href=')[1].split('>')[0]).replace ("\'"," ")

                    ident=ident.split('=id')[1]
                    a=len(ident.strip())


                    ident= ident[0:a-1]
                    CoordenadaX= m.group(1).split('<a')[0].split(',')[1]
                    CoordenadaY = m.group(1).split('<a')[0].split(',')[2]
                    archivo.writelines(" ide : " + ident + "\n")
                    archivo.writelines("CoordenadaX : " + m.group(1).split('<a')[0].split(',')[1]+
"\n")
                    archivo.writelines("CoordenadaY : " + m.group(1).split('<a')[0].split(',')[2]+
"\n")

                    if "dal</strong" in listas:
                        pos =listas.index("dal</strong")
                        #print s.group(1).split('>')[pos+1].split('<')[0]
                        FechaDesde=s.group(1).split('>')[pos+1].split('<')[0]
```

```
                    archivo.writelines("Fecha Desde : " +
s.group(1).split('>')[pos+1].split('<')[0]+ "\n")

                if "al</strong" in listas:
                        pos =listas.index("al</strong")
                        #print s.group(1).split('>')[pos+1].split('<')[0]
                        FechaHasta=s.group(1).split('>')[pos+1].split('<')[0]
                        archivo.writelines("Fecha Hasta : " +
s.group(1).split('>')[pos+1].split('<')[0]+ "\n")

                if "Sede:</strong" in listas:
                            pos =listas.index("Sede:</strong")
                            #print s.group(1).split('>')[pos+1].split('<')[0]
                            Lugar=s.group(1).split('>')[pos+1].split('<')[0]
                            archivo.writelines("Lugar : " +
s.group(1).split('>')[pos+1].split('<')[0]+ "\n")

                if "Indirizzi:</strong" in listas:
                                pos =listas.index("Indirizzi:</strong")
                                #print s.group(1).split('>')[pos+1].split('<')[0]
                                Direccion=s.group(1).split('>')[pos+1].split('<')[0]
                                archivo.writelines("Dirección : " +
s.group(1).split('>')[pos+1].split('<')[0]+ "\n")

                Eventos=
Salida(ident.strip(),CoordenadaX.strip(),CoordenadaY.strip(),FechaDesde.strip(),Fecha
Hasta.strip(),Lugar.strip(),Direccion.strip())
                serialize = json.dumps(Eventos, cls=UserEncoder, indent=4)
                serializeTotal =serializeTotal + serialize

                break
        dia+=1
    mes+=1
archivo.close()
archivoJSON = open("c:\\personal\\actividadesculturales.JSON", "w+")
archivoJSON.writelines(serializeTotal)
archivoJSON.close()
print "FIN"
```